The ACMS, a model-driven approach to new interactive television services production and delivery

Roberto Del Pero
Rai RadioTelevisione Italiana-CRIT
C.so Giambone, 68
10135 Torino
+39 011 8103251
r.del_pero@rai.it

Fulvio Negro
Rai RadioTelevisione Italiana-CRIT
C.so Giambone, 68
10135 Torino
+39 011 8103229
f.negro@rai.it

Luca Vignaroli
Rai RadioTelevisione Italiana-CRIT
C.so Giambone, 68
10135 Torino
+39 011 8103128
l.vignaroli@rai.it

ABSTRACT
This paper contains a brief report of the analysis of new requirements raised by Interactive Television and then gives a possible solution by defining a model-driven environment for the production and delivery of interactive television applications and their content.

Categories and Subject Descriptors
J.7 [**Computers in other systems**]: *Process Control, Publishing.*
H3.7 [**Information storage and retrieval**]: *Digital libraries – User issues, Systems issues.*

General terms
Management, Design, Human factors.

Keywords
Interactive Television, Content Management, Application management, Content publishing, Television production.

1. INTRODUCTION
New interactive television technology is having a major impact on industries producing applications and their content. Consumer expectations are extending beyond conventional radio and television.
Cross-functional and configurable content, referred to as smart media, is gracefully but irreversibly substituting traditional formats. Eventually, mechanized and sequential manufacturing methods must be replaced by scalable, configurable and agile methods that can support a spectrum of delivery platforms and adapt the content to conform to end user's required area of interest.

Rai's production experience as well as its deployment of new interactive applications for Digital Terrestrial Television (DTT) has raised new requirements which are relevant in any interactive television scenario.

This paper contains a brief report of the analysis of these new requirements and then gives a possible solution by defining a model-driven environment for the production and delivery of interactive television applications and their content.

The Application and Content Management System (ACMS) is the solution designed by Rai Research Centre in Turin to cover new requirements of interactive television scenarios.

2. STATE OF THE ART
Classically, a *Content Management System (CMS)* [1-4] is a software application deployed primarily for interactive use by a potentially large number of contributors. The managed content typically includes computer files, image media, audio files, electronic documents and web content. A Content Management System is most often used as an archive as well. CMS systems are also used to store files in a non-proprietary form and to share files easily in their data communication infrastructure as most systems use server-based software. Many Content Management Systems include a feature for Web content and some of them have a feature for a "workflow process" [5].
The idea behind workflow management is to move an electronic document along a process chain for either approval or for adding content. Some Content Management Systems will easily facilitate this process with email notification and automated routing. This is ideally a collaborative creation of documents. A CMS facilitates the organization, control and publication of a large body of documents and other content, such as images and multimedia resources.

A *Web content management system* is a content management system with additional features to support tasks required to publish web content to web sites. Web content management systems are used for storing, controlling, versioning and publishing industry-specific

AMDIT 2008, February 14, Quebec, Canada

DOI 10.4108/ICST.AMBISYS2008.2843

Channels \ Terminals		Television			Mobile			PC
		STB without HDD	Advanced STB with HDD	Media Center	Phone	PDA	Car TV	Workstation
Broadcast	DVB (T, T2, S, S2, C)[11]	SD/HD TV	SD/HD TV, PVR	SD/HD TV, PVR	N/A	N/A	N/A	TV on PC
	DVB-H[13-14]	N/A	N/A	N/A	Mobile TV	Mobile TV, PVR	Mobile TV	N/A
	Data-broadcasting	N/A	Push TV	Push TV	N/A	N/A	N/A	Push TV
IP Connection	Streaming RealTime (Mcast-Ucast)	N/A	N/A	TV Simulcast	Network Expensive	Network Expensive	Network Expensive	TV Simulcast
	Streaming VoD	HD/SD IPTV	HD/SD IPTV	HD/SD IPTV	Network Expensive	Network Expensive	Network Expensive	Web TV VoD
	Download unicast	N/A	A/V pod-cast, P2P	A/V pod-cast, P2P	A/V pod-cast	A/V pod-cast	N/A	A/V pod-cast, P2P
	Download multicast	N/A	Push TV	Push TV	N/A	N/A	N/A	Push TV

Table 1: Services matrix of Channels vs. Terminals

documentation such as news articles, operators' manuals, technical manuals, sales guides and marketing brochures. A content management system may support several features like import and creation of documents and multimedia material, identification of key users and their content management roles and the ability to assign roles and responsibilities to different content categories or types. It can additionally define content workflow tasks, including the ability to track and manage multiple versions of a single instance of content and the ability to publish the content to a repository for content access support.

We can see the *Interactive Television* as a system with layered functionalities, from low interactivity (TV on/off, volume, changing channels) to moderate interactivity (simple movies on demand without player controls) and high interactivity in which, for example, an audience may affect the evolution of the program being watched. The most obvious example of this would be any kind of real-time voting on the screen, in which audience votes create decisions that are reflected in how the show continues. In general a return path to the program provider is not always necessary to have an interactive television experience.

Many existing technologies permit the development of Interactive Television applications; with the launch of the *IPTV (Internet Protocol Television)* [6,7], Television and Web share a large set of techniques to allow the final user to have an interactive experience in front of the television set. Quite obviously, every week, a new family of technologies enters into the world of TV application development [8].

The most used technologies employed today for *Set Top Box* (STB - client user device) oriented applications are the simple *Video Overlay* or *Multimedia Home Platform (DVB-MHP)* [9] and *MHEG-5 (Multimedia and Hypermedia Experts Group – Part 5)* [10]. Applications developed with MHP and MHEG are in most cases delivered directly through the transmission channel. The choice of the use of these technologies by the service provider depends on the middleware and the application layer that is available on the client user device.

MHP is an open middleware system standard designed by the DVB[11] project for interactive digital television. The MHP enables the reception and execution of interactive, Java-based applications on a TV-set. Interactive TV applications can be delivered over the broadcast channel, together with audio and video streams.

MHEG-5 is an object-based declarative programming language which can be used to describe a presentation of text, images and video. An MHEG-5 application consists of a number of *scenes* which the user of the application can move between. Each *scene* lists the items of text and graphics to be presented and can contain blocks of procedural code which are executed in response to one of a predefined set of events such as keys being pressed, timers firing or content being successfully loaded into memory.

For IPTV Application developing, many integrated tools are the same used for Web applications, like *HTML*, *JSP*, *Ajax*, *Silverlight* [12]. In most cases of IPTV Applications the return channel is mandatory and the application is exposed like a web site from an application server.

Many channel technologies are today available for TV application delivery. The Table 1 shows the Services matrix which defines an interactive Service as intersection between the Channel on which the Application or Content is

transmitted and the target Terminal on which the application or content is shown to the final user.

At the same time, another distinctive element for Interactive Television is the *back channel* typology and again many return paths could be involved in the developing and in the approach of the interaction, like Fixed or Mobile, always-on or not.

3. ON-FIELD REQUIREMENTS

The Rai experience on production and deploying of interactive applications for DTT television has raised new requirements which are relevant in any interactive television scenario.

Many actors are involved in the overall workflow for the production, deploying and maintenance of an interactive television application. They can be grouped in three main layers: *Development*, *Technical* and *Editorial*. A useful strategy in order to optimize the process is making competences independent each other though at the same time actors shall be working in a cooperative way.
It is necessary to find an environment which enables to merge these different approaches, taking into account each competence of each actor involved in the management workflow.

Applications shall be grouped in order to follow editorial concepts taking into account the fact that at the same time each application is different from a physical point of view.
This environment shall allow the developers staff to translate the complexity and the internal structure of the application in order to allow other actors to work easily on the application, each from a different responsibility point of view.

When the application is ready for publication, all information needed to go on line and to correctly follow the expected editorial behavior has to be available. Before and during the on line period, the application needs to be provided with multimedia contents, therefore the editorial staff should be able to setup the application without knowing the complexity of the physical version of the application itself. At the same time, the technical staff responsible for publication should be able to configure the publication and scheduling parameters without knowing the internal structure of the application.

This environment shall allow the publication to different target channels/terminals. It will focus its efforts to implementation of the architectural and data model in the part of media delivery and user terminals taking in to account the following aspects:

- Automation
- Scheduling
- On Air – On Line Planning
- Scalable distribution
- Monitoring

4. THE RAI APPROACH

The approach adopted by Rai provides for two different types of main context and it takes into account application and content publication functionalities.

4.1 Version Context

The Version Context (Figure 1) allows actors to save and to access to *applications*, *application components* and *aggregation of application* (*Programs*) according to a version logic: from the editorial point of view, each element is inserted in a context of structure made of Project, Editorial Entity, Editorial Version and Physical Version.

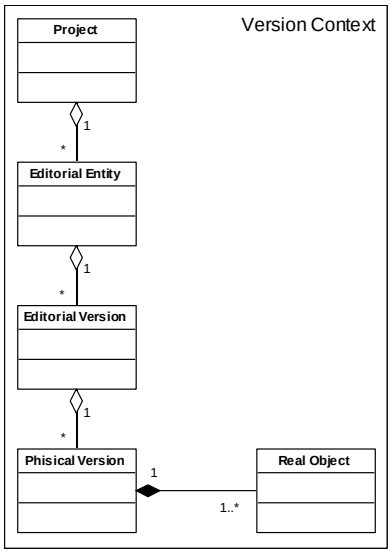

Figure 1: Version Context scheme

The *Project* element of the structure allows actors to manage sets useful for the cataloguing and the aggregation of heterogeneous elements which refer to a particular editorial/functional project.

The *Editorial Entity* is the aggregate which collects and puts in relation various publishing versions concerning a same publishing concept; this editorial entity is the collective identification of all editorial versions it aggregates. This can be useful to report a set of editorial versions without specifying a particular one. A reference to an Editorial Entity can be useful during the service planning when the precise identification of a particular editorial version is not necessary or is not possible.

The *Editorial Version* is a version of an editorial concept which is specifically characterized by determinate information of editorial type and content. Typically this concerns the informative, artistic and communicative

dimensions, as well as the attribution of credits and other production information. The Editorial Version is the main object entity of the ideation, editorial definition and all decision processes which determine the content of the work; the editorial version shall be considered as the concrete landing point of all processes which participate in the building of the editorial shape. The information which characterizes an editorial version is independent from the physical versions which realize it.

The *Physical Version* is a concrete mode through which a particular Editorial Version is realized; it is characterized according to various dimensions, one for each of its possible components (audio & video, static images, text, graphics, interactive applications). The structure of a physical version may be considered as the "multimedia view" of a certain editorial version. The multimedia view is the main object of the editorial ideation operations of an application.

4.2 Composition Context

The Composition context (Figure 2) keeps in account the composition relations between the various entities.

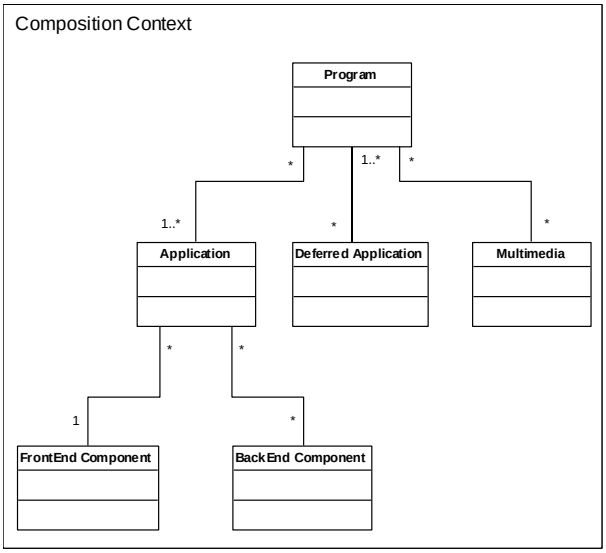

Figure 2: Composition Context scheme

The *Program*, at the highest level, represents the total publishing entity which the final object to be published is derived from. Under the composition point of view a particular program contains at least one application and possibly one or more multimedia contents; at program level there is also the possibility of making some fictitious links towards applications which will be actually chosen later on (deferred applications).

Each *Application*, in its turn, is composed by a single Front End Component and may hold several Back End Components. The Application represents the publishable atomic object.

The physical version of the *Front End Component* is the main entity of the application management system; it is the set of all the files concerning the application tree of the application, together with all the other files that are used for the real physical building of the application itself. So, the physical version contains the references to all the physical files which constitute the concrete entity.

Each *Back End Component* can be used in more than one application, as well as every application can be contained in more than one program.

It is necessary that the various applications which compete to the formation of a program are linked to the program itself in an ordered way to guarantee the possibility of linking to the program the same application several times, every time characterized by various parameters or data; each member relation is identifiable in an univocal way with respect to the others. This identification of composition relationship is maintained consistent in every section of the version context: the first relation in the editorial entity section is equivalent to the first relation in the editorial version section concerning the same product and so on.

4.3 Delivery of Applications and contents

As depicted in Figure 3, the solution foresees that a set of steps will be performed in order to publish an application.

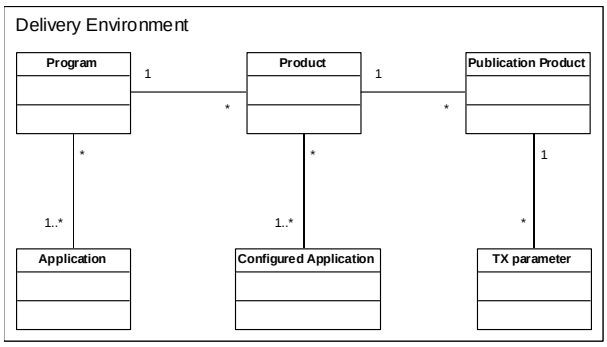

Figure 3: Delivery Environment scheme

The delivery logical flow starts from the *Application*; one or more applications could be linked to form a *Program*, from which the *Product* is built by resolving links to all parameters and static contents. The Product is defined as the object that contains the application and all related contents and configurations (*Configured Application*). In the logical flow after the Product there is the *Publication Product* which is built, starting from the Product, defining transmission parameters and linking the dynamic contents.

The solution foresees that content incoming from an editorial staff is redirect to a unique *Content Publication Site* in order to have a shared location in which content shall be easily discovered and addressable as an object hierarchy.
The *Content Publication Site* is independent from any Content Management System used by a stand alone editorial staff. Each editorial staff can decide the tool to use for the content creation and the workflow to use for the content validation. For each Editorial staff is mandatory to publish content in the assigned *Content Publication Site Area*; each Editorial staff will create their own object hierarchy in the publication site.
The editorial staff should publish on the *Content Publication Site* the physical file or some procedures which recover content files from proper repositories (e.g. database) and make themselves available for the publication. Contents on the publication site should be shared by several user applications.

Inside the Content Publication Site a uniquely addressable object is called *Content Source*. Content Sources are part of the input for the final publication engine which is able to manage each Content Source performing, if needed, a *Transformation Action* and it is able to deploy the real content in a given entry point of a particular application (*Destination*). The triplet *CS-T-D* (*Content Source–Transformation–Destination*) is the core of the *Publication Engine* that knows actually deployed applications and which is able to deliver content on each of them.

When the Product is built a set of triplets CS-T-D is linked to each application to be deployed so that the application and its related content will be automatically managed during on line time. When the application is on line many actors perform specific tasks to manage dynamic contents: the editorial staff makes available contents on the Content Publication Site, with regular or variable frequency according to a particular *Editorial Scheduling*; the application, with a determinate frequency, makes the refresh of its own dynamic contents, without responsibility about the presence and the accuracy of the contents themselves (*Application Scheduling*); another scheduler, finally, reads the source and the destination of every content from the application content structure and it moves the content itself from the Source to the expected Destination after the related Transformation Action; these scheduler operations are made with a smart frequency according to Editorial and Application Scheduling (*Content Scheduling*).

5. THE ACMS

The *ACMS (Application and Content Management System)* is the approach based on the previous analysis designed by Rai Research Centre in Turin to give an answer to new requirements of interactive television scenarios.

The ACMS allows all involved actors to cooperate to publish applications, opportunely set up and fed with contents, on the *transmission system* according to timings established by transmissions scheduling system. Such applications are made available on the *user devices* and allow to implement complex services which require, potentially, the use of a return channel between users and a service centre. The ACMS interacts with the *return channel* to allow the realization of such interactive services; it can interact with the *authoring systems* on which the applications are developed and with external *sources of contents*.

The ACMS allows users to manage the applications with related contents and their corresponding histories; it also allows the management of products used for various publications and the management of their history, in addition to provide functionality of logging and workflow management.

The Figure 4 shows the internal architecture of the ACMS.

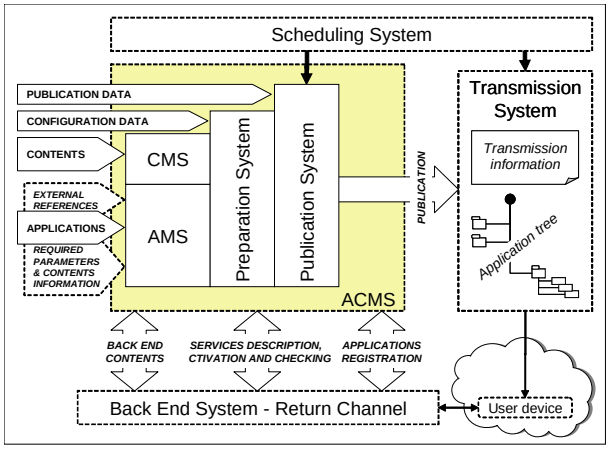

Figure 4: ACMS subsystems

5.1 The AMS

The *AMS* is the subsystem which allows users to manage all kinds of application, loading them on the system and making them available for the configuration and subsequent publication.
The AMS makes available the management of applications and all related components (e.g. Front end and back end components) ensuring their persistence saving physical files which contain the code of each application components. Two different components are managed from the AMS: the Front End component and the Back End one. The first one consists of the application part designed to be executed on the user device; this component is structured in directory trees and can be physically saved, for instance, as zip files containing the whole application tree. The Back end component is the part of the application which runs on the

server side as *service*. The AMS provides the functionalities that allow the insertion, the management, the extraction and the deletion of every single application component as atomic entity; for each application the system permits the set up of a particular workflow management, depending on the membership of such element to a particular kind of service (e.g. interactive advertising).
The AMS allows users to access the single application according to the composition context and the version context (Figure 5); in addition, it allows users to handle accessory information (metadata) useful to the characterization of the various objects in the system. In fact every kind of entity inside the ACMS is characterized by the presence of information concerning the use, the identification and the description of a particular item belonging to such typology.
The AMS allows users to manage the single context entities and their own attributes in creation, modification and deletions; in order to assure the possibility of restoring the previous contents after modifications or deletions (rollback), the system guarantees an internal versioning for every single entity (Figure 5).

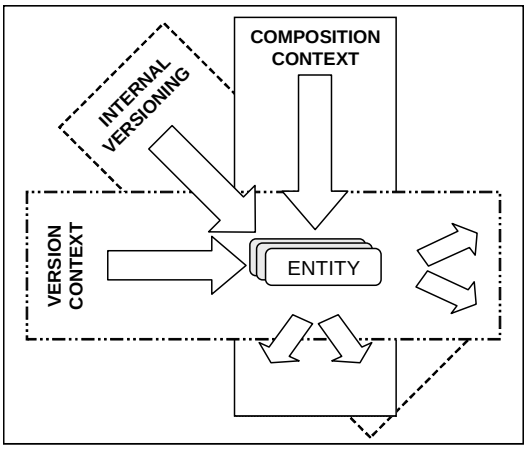

Figure 5: ACMS - Managed context

The AMS includes among its functionalities the management, for every single application, of all information concerning parameters, contents (structures, definitions, default values and transformation proceedings) and the external references. This information will be used during the configuration and publication phases.

5.2 The CMS

The CMS is the subsystem which manages the various kinds of contents, ensuring their persistence and availability for the different ways of publication: the CMS subsystem permits to make contents persistent for future reuses both on the same type of service (same contents used by more applications of the same type) and on different services (same contents which feed different type of applications, e.g. WEB pages and DTT applications).
Contents in the CMS are handled according to a process logic which implements the typical functionalities of a Content Management System: feeding, integration and assembling, transformation, editing, retrieval and exportation; all these functionalities are useful for the following use of contents in publication mode and they are controlled by a workflow management system.

CMS feeding can occur by different ways. Contents may arrive from external flows and they are made available through external feeds everyone with its own structure: in this case the system works both in *pull* mode (CMS asks the source for new contents and upload them) and in *push* mode (the external source of contents signals to the CMS the presence of new content items and starts up the uploading). Another way to feed the CMS is to use some interface for data insertion: contents are uploaded in the CMS exploiting proper GUI that works directly on data structures in the system.
Integration/assembling functionalities permit the creation of new content items starting from other contents both internal to the CMS and provided by the outside (e.g. CMS makes available the merging of contents provided by an external source with contents inserted ad hoc through a GUI to achieve the real content to publish); the system ensures the persistence of such integrated/assembled contents either saving the new content or saving the assembling rules and the building blocks.
Transformation functionalities permit to obtain new contents derived from other present in CMS through operations of manipulation that go beyond a simple assembling and the system must guarantee the persistence of such contents.
Editing functionalities allow users to change the contents inside the CMS using a dedicated GUI; during these actions the system ensures versioning, rollback and approval functionalities.
The *retrieval functionalities* of CMS contents is done basically through the use of identification and description information (e.g. Title and description field), while more complex identification methods, content based, are exclusively implemented for particular kinds of contents.
Through the use of *export functionalities*, it is possible to make contents available for Preparation and Publication subsystems.

There are different actors with different responsibilities working on CMS; they contribute to the management of contents from their insertion in the system until their approval and export. The workflow logic depends on the sort of analyzed content and on the final use established for the content itself. The CMS allows the generic contents

automatic management shared among all applications in the ACMS.

In the context of the CMS, Editorial Procedures are defined as a subset of functionalities provided by the CMS, they are coordinated by an appropriate workflow and they operate on a particular set of data. The building of an editorial procedure on the CMS permits to define an environment in which content providers work following specific steps and interfaces. Generally the system works with two different kinds of Editorial Procedures: those directed to data allow users to insert and to manage contents not necessarily connected to a particular application (e.g. an Editorial Procedure which introduces the stock exchange data in the system). Otherwise the application oriented Editorial Procedures are built to map the relation between a specific application and related contents (e.g. an Editorial Procedure for the staff which introduces the graphics of TV Guide application home page in the system). This type of Editorial Procedure operates on data structures directly connected to the internal data structure of the application it refers to. The CMS makes available the creation, the management and the use of generic Editorial Procedures.

5.3 The Preparation system

The Preparation system is the module of the ACMS which manages the Product entity which is created starting from data coming from AMS (application and descriptive part) and CMS (contents part) joined together with configuration data provided if necessary during the creation of the entity itself. The Product holds all static components needful for publication, properly set up and updated.
Inside the Preparation system is possible to create new products starting from appropriate programs, connecting all expected applications. For particular kinds of programs during the preparation phase the publisher has to decide which are the real components to publish (deferred applications). Furthermore in this phase the publisher makes the configuration of such Product available both from functional and transmission point of view: the subsystem allows users to give values to parameters useful to define the behavior and the transmission mode of every single application involved in a particular program.

The effective contents used by the single application are inserted in the Product, extracting them from the CMS or directly introducing them through a specific GUI; this operation is possible when a static feeding mode was chosen, otherwise, if a deferred static or dynamic feeding mode was chosen only references to contents used by single components are inserted in the Product. Such Product is now ready for publication and it is made persistent in AMS as further entity of the structure: Editorial Entity, Editorial Version, Physical Version and Product.
The Preparation system is able to work on already defined Products, originating new products from them simply changing characteristics parameters and/or contents (real contents or references to contents). If Products are not published yet it is possible to make changes (following particular workflows) without having to define a new Product; in this case the system guarantees versioning and rollback functionalities. If the product has already been published it is not possible under no circumstances to introduce changes (integrity of the product).

There are many actors with different responsibilities working on Preparation subsystem, everyone contributing to the Products management from the creation until the publication. The workflow logic may depend on the type of product.

5.4 The Publication System

The Publication System produces the physical object to publish, called Publication Product, which is built starting from a specific Product. The Publication System is also able to publish Publication Products under the scheduler control. During the publication phase, contents managed by dynamic feeding mode are connected; in addition the Publication Product contains information concerning publication scheduling (e.g. expected/real publication time, the service to publish on). The new entity is made persistent in AMS as further entity of the structure: Editorial Entity, Editorial Version, Physical Version, Product and Publication Product.
Regarding the front end components, the publication system copies the physical object on broadcast channel building and bringing up to date the application trees concerning the applications which compose the program. At the same time the publication system sends to the broadcast channel all transmission information needful both for manage the real transmission towards user devices and for give them possible parameters used by application during execution time.

The updating of application trees is the mode used to manage the dynamic contents (contents which change during the execution of the application). The system guarantees two different modes of updating of the application tree: the complete updating mode where, apart from changes inside the application tree, the complete tree is moved towards the broadcast channel, or the selective updating mode in which just files which have had a change are moved.
It is foreseen that the updating strategies are performed in polling mode, in which the publication system manages changes according to scheduling and in push mode, in which the system brings up to date contents that have to be updated as soon as changes happen.

In general when a change in a dynamic content takes place the system has to notice the happened change, find the content from the CMS, execute the transformation process

associated to the datum, work out again the application tree associated to the application and finally publish the up to date application.

6. CONCLUSIONS

The production and the management of new interactive television services raise new requirements not covered by actual CMS systems.

A clear division between competences and actions that work together in the production of new interactive television services is mandatory. The new generation of production and management systems has to guarantee coherent views for each element in each context.
In order to optimize the management and the reuse of content elements, a CMS that exposes its own content elements through a shared and well known interface is recommended.
All applications which compose new services cannot be managed directly by a common CMS because they need new and not yet implemented features. Therefore a new type of integrated management system is required (Application Management System).
The delivered service is made of configured applications which refer to related contents. Models and Systems, using the CMS and the AMS outputs, are required to automatically integrate applications and contents.

The solution described in this paper give a model-driven environment for the production and delivery of interactive television applications and their content and it is the base for the development of the Application and Content Management System, the solution designed by Rai Research Centre in Turin to cover new requirements of interactive television scenarios.

7. ACKNOWLEDGMENTS

Authors wish to thank Alberto Messina and Mauro Rossini for their precious support.

8. REFERENCES

[1] J. Heitmann - *Content Management System for television production* - EBU Technical Review - Summer 1999

[2] J. Heitmann - *Future program production: implementing enterprise wide content management systems - Same actual case studies* - IBC2003 Conference Publication

[3] J. Robertson - *Centralised or Decentralised authoring?* - KM Column - April 2002

[4] J. Robertson - *11 usability principles for CMS products* - KM Column - May 2007

[5] Workflow Management Coalition Document Number TC00-1003 (1995 – 01): *The Workflow Reference Model*

[6] International Telecommunication Union - ITU-T IPTV global technical Workshop - Driving the future of IPTV - Seoul, 12-13 October 2006 – *Outcome Documents*

[7] Franc Kozamernik – *IPTV, a different television?* – EBU DIFFUSION online - 2006/30

[8] Ian Kingsford-Smith - *Modelling the business of enhanced television* - Broadcast Asia, 2005 International Conference

[9] ETSI Document TS 102 812 (2001 - 11): *Digital Video Broadcasting (DVB); Multimedia HomePlatform (MHP) Specification 1.1*

[10] ISO/IEC 13522-5:1997 Information technology - Coding of multimedia and hypermedia information - Part 5: Support for base-level interactive applications

[11] Ulrich Reimers -*DVB, The Family of International Standards for Digital Video Broadcasting*, Second Edition, 2005, ISBN 3-540-43545-X

[12] P. Kafno - *Authoring for the interactive content generation* - IBC2007 Conference Publication

[13] ETSI Document EN 302 304 V1.1.1 (2004 - 11): *Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H)*

[14] ETSI Document TR 102 377 V1.1.1 (2005-02): Digital Video Broadcasting (DVB); DVB-H Implementation Guideline