A Software Framework to alleviate the Development of DR Management Systems

Madhavi Latha Yalamarthi, Madhuri Katabathuni, Jyothi Kameswari Uppuluri

Abstract— Even though many of the publications sustain the use for a generic DR Management software architecture, we noticed that present attempt to describe a Digital Rights Management Architecture that it doesn't provide to able support to visualize the existence and management of DR Management systems and applications. It is a marked problem that denote an essential demand for the development of DR Management, provides the significance of software architecture on the functional and non-functional characteristics of the performance. In this paper we suggested a generic DR Management framework, figure out it in the contingency even in heterogeneous environment. Analyze it to relevant work in the Digital Media Project. The proposed framework is more informative than the relevant work observed so far

Index Terms— DR Management, DM Project, Component classes, Illustration, Analasys, Mediators.

1 Introduction

NOW A DAYS Software framework technology plays a significant role in the advancement of DR Management systems. The DR Management association has approved the significance of generic software that directs developers and that empower the rework of third party components. Flashing on present form of art DR management systems, we noticed three standard aspects [1]. Primarily, Most of the DR Management systems perform a general set of essential services regarding content handling, authentication and access control. In previous works, we have observed the main DR Management systems, and finding the specific components. Also, Contemporarily DR Management Systems expand the specific functionality with a large-scale of services, similarly consumer tracing, payment or fault recognition. And finally, the overall scope of DR Management is speedily advancing regarding security technologies, authentication policies and payment systems.

With this information elevate three valuable demands that have impression on consumers, inventors and promoters of DR Management secured content. Primarily, there is demand for a systematize specific framework that can be rework by content promoters to promote affiliation of DR Management in their content delivery platforms. Also, an enhanced framework, which presents DR Management services apart from the specific functionality, should allow diverged points because to assist diverged requirements of content consumers, inventors and promoters. Finally, given the speedily advancing character of DR Management, even in heterogeneous environment.

The contribution of this work is to present the DR Management Software Framework that cope with above demands.

The framework has been foster in the contingency of a research project which combines consumers, inventors and promoters [2].

It is based on contemporarily framework patterns which make the framework very easier to perceive, enhance and customize. This work figures out the framework and analyzes it with ideas from the DM Project [3].

The remaining of this work is designed as follows. Part [1] [2] chooses and describes the main quality specifications of the DR Management system. It gives a complete picture of three key fundamental collection and dive in on the framework from three aspects content handling, consumer tracing authentication and access control. Part [] assess the suggested framework and analyze it to related work in the DM project [1] [2].

2 Prerequisites

This part chooses and determines the main quality aspects that are related to DR Management systems. A quality aspect is a resource of a framework by which the consumers can decide its quality. These characteristics, like for example perceive, enhance and Customize may have specific effect on the frame work of a system [4]. Similarly, quality aspects are normally approved to be desperate points over the prior stages of system development. It must be crystal clear that essential software aspects should be supported when developing and deploying DR management systems.

It is crystal clear that commonly software quality aspects must be encouraged when developing and deploying DR Management systems content allocation applications: interoperability,enhancebility,customize,applicability,analysis,execution,cer tainity,security,etc.In this paper we conclude our analysis to the three key points: interoperability, enhancebilty, alterability. The reason for focusing these is speedily advancing DR Management area. Interoperability capability of two or more attributes from different marketers to associate at execution. Interoperability between DR Management systems is awfully essential time given the intricacy and broadness of DR Man-

- *Madhavi Latha Yalamarthi is currently working as a Assistant Professor in Department of ECM in K. L. Univresity, India, PH-+91-8897040490. E-mail: s.madhavi.y@gmail.com*
- *Madhuri Katabathuni is currently working as a Assistant Professor in Department of ECM in K. L. University, India, PH-+91-9885928333. E-mail:madhuri.katabathuni@yahoo.com*
- *Jyothi Kameswari Uppuluri is currently persuing Ph.D and working as a Assistant Professor in Department of ECM in K. L. University, India, PH- +91-9949623942. E-mail: krovi_jyothi@yahoo.co.in*

agement operation as well as speedily advancing area, which denote different alternatives of the same service and be able to coordinate. Significantly, customers must be able to receive and take in content and privileges from any content promoter. Inventors should provide content and powers only once and be able to foster between different promoters to appropriate their content. Promoters must be able to restore utility components with analogous components from various marketers.

2.1 Alterability

Alterability is the efficiency with which a system can accord adjustments to its software. The performance of the publishing system must be alterable to accord differing business codes for powers, payment and rights authority. In the way that, a promoter must be able to alter its payment model from pay-per-view to a monthly supplement. The customer should be able to alter the action of their DR Management client in spite to submit anonymous depletion data contrary to privacy sensitive data.

2.2 Enhancebility

Enhancebility is the ease to include new elements to the system. Hence the section of software security and DR Management is awfully dynamic; DR Management systems must grab change and be open to enhance their working with developed components. Customers and promoters, comparatively should support their system up-to date by dynamically summate security updates.

3 Framework

Based on these specifications is not a minor task to accomplish. This part primarily discussed an overview of DR Management architecture by recognizing three consistent combinations of components and describing the interaction between them. It also shows how two important points can be addressed by employing framework patterns. Also, it proposes the important components by flash in on the framework from three views: content handling, authentication and access control. Specifications about component interfaces and illustrative use cases have been specified in a technical report. [5].

3.1 Component Classes

A component class presents a consistent collection of components that offers an important functional view of DR Management. We recognized three component groups (1) DR Management services, (2) content illustration and analysis and (IA), and (3) security algorithms. Figure illustrates the relationships between component groups at customer, promoter and inventor side. Hence a DR Management framework is embedded within a publishing context, functionalities assign to user, content or input administration can frequently be rework.

3.1.1 Utility Component Class

This component class accords the main utilities of a DR Management system: content illustration, authentication and input administration. A few of these utilities can be rework from the content administration platform at the promoter side.

3.1.2 Illustration and Analysis Component Class

This component class is used to denote content and privileges, to collect data about content usage, and to analyze privileges.

3.1.3 Authentication Component Class

This component class suggests various authentication algorithms to provide security for the content against damage by encrypting and damaging. These components are used by illustration and analysis group. For example, to impose authentication policy. The Framework uses the following two important issues. Firstly, it reinforces translucent message interference to support customer tracing. In order to allow customer tracing at the promoter side, customer requests should be interfered by the promoting system well before they are taken by the promoting system component group. At the customer side, customer requests must be interfered prior to attaining the customer security component group. Significantly, the framework must enable message interference at different sites. Moreover, customer tracing is a non-functional DR Management affair that may not regularly be present, message interference should be translucent for service components.

Also, since content may not regularly be executed in first pass over the security component class, the framework must facilitate periodic execution in a manageable way. It is uncommon to design component classes as a layered system continuous interaction flow [6]. Content may contain various parts, each part can have different privilege affiliated with it resulting in various ways of security, for example, the denominations of a video stream in various languages perk tracks that can played after a specific date. The IA component class comprehends knows how framed content is illustrated and significantly how these parts are united together and guarded. The security component class, though, cannot differentiate various content components with different security specifications.

The two above mentioned issues, the use for transparent customer tracking and non-consecutive content traversal over the system can be clarified at the framework level by applying the intermediate framework style [7]. A framework style conveys a framework structural alignment schema for software systems. It offers a set of predefined sub-systems specifies their liabilities, and combines rules and guidelines for constituting the relationships among them.

Three framework patterns are needed for clearly intercepting messages and transmitting them in different ways: Router, Mediator and Blackboard (see also Figure 1).

A General attribute shared by these patterns is that they confirm that components never communicate directly together, but always through a coupler. The coupler is called router, Mediator or Blackboard, based on the performance it plays in the pattern. In the router arrangement, source component starts communication that is routed to any of the multiple destination components. The choice of the destination component is managed by distribution rules. The mediator expands the router arrangement with the capability to break down and reorganize incoming messages. Moderate results are stored in work in progress component. In the blackboard arrangement, various components use shared date structure which combines state data and activates components whenever state

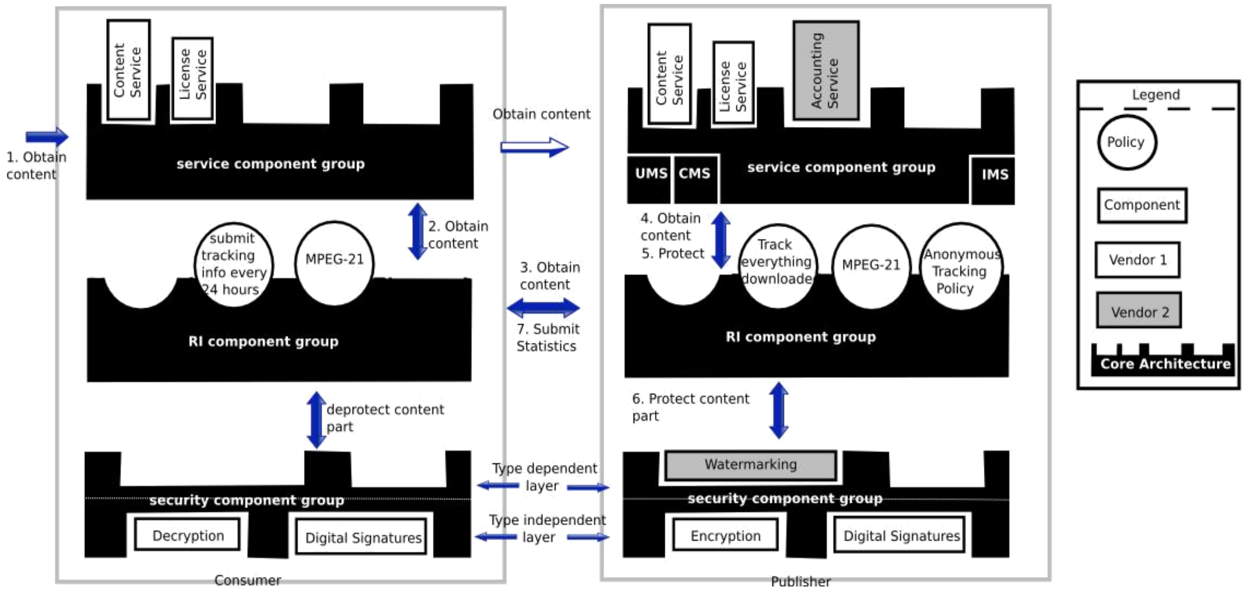

Fig. 1. Framework overview that describes two important roles, customer and promoter, both consists of three component classes. The service component class contains main promoting servicing and pluggable DR Management service components. The grey and white components show services supported by various businessman which describes interoperability between service components. The IA component class is used to describe content and privileges analyze business strategies to change the systems attitude it works as a dispatcher that dispatch request according to, for example a customer tracing strategy. The security component class supports components with algorithm for encryption and decryption. The system can be enhanced by including components and strategies.

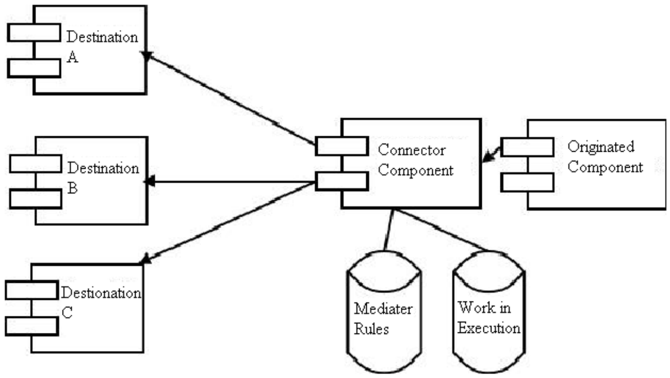

Fig. 2. The router, mediator and blackboard framework patterns are used for clearly blocking messages and delivering them in different ways the originated component initiates communication and are forwarded to zero one or more manageable by distribution rules (A, B, C). Selection of the destination component is manageable by distribution rules in the connector component and intermediate results are stored in a work in execution component.

changes. State data has been changed in the blackboard by respective components, thereby increasingly providing a solution to the problem stake.

The selection for the mediator arrangement is validated for the following reasons: (i) tracing is feasible, because interference and alterability of message supported, (ii) novel components can be easily added and (iii) the mediator can employ different business rules, such as dynamically altering content or privileges for various platforms (e.g. cell phones and pc's). we should care that the mediator can become a barrier and causes failure because it interfere all communication. However the black board has more defects, because all components must use the similar representation which is difficult to alter data depictions.

3.2 Frame Work Views

Now, flashing on the Frame work from three views. The content handling flashes on the base components of DR Management systems, the customer tracing view shows how the framework can be expanded with noticing functionality and the security view finishes the content promoting loop explaining how content can be protected by different approaches.

Each view can be viewed as a filter that flashes on a specific level of functionality. The bottom level suggests the base functionality required for delivering content. The remaining levels expand this functionality with tracing and security which collectively forms a framework which encourages inception, distribution and usage of DR Management protected content, distribution of privileges, and customer tracing. For Every view the base components are summarily introduced.

3.2.1 Content Handling View

This view depicts all services regarding content distribution and handling. Compatible components are situated in the service and the IA component class.

3.2.2 Service Component Class

In earlier work to recognize seven key DR Management service components for privileges, access, content Handling, damager identification, content gathering, customer tracking and payment. DR Management implementers must not reformulate the wheel and rework existing components whenever possible. For Example, a promoting framework typically procures support for content handling, access and payment. Significantly, components such as content service, the user management system and the charging system can be easily reapply possibly by accepting some strategies.

3.2.3 IA Component Class

The IA component class contains a representation component, which is authoritative for generating content in a particular representation and for altering content in to another representation.for example a representation from MPEG-2 to MPEG-21 [8].

3.2.4 Tracing View

Usage stats can be acquired from the promoter side or from the customer side. The depicted solution is same for both sides. The following components and IA component class offer support for customer tracing.

3.2.5 Service Component Class

The accounting service uses the data that the enrolling component has gathered in order to provide outlines of statistical data, e.g to provide monthly utilization bill.

3.2.6 IA Component Class

The dispatcher framework arrangement can be imposed to perform the functionalities of this component class using the following components.

3.2.7 Strategy Engine

Resolve strategies depicts. At the customer side, for example, a strategy represents the privilege described in the MPEG 21 REL.

3.2.8 Dispatcher Component

Delivers the message to the proper component class.

3.2.9 Enrolling Component

Enrolls and delivers proper messages to the tracing component.

3.2.10 Representation Component

This component has been described above. Each internal IA component records with the dispatcher component and introduce the dispatcher component with the impressed messages.

The Dispatcher Component refers to a list of delivery policies in policy machine to choose how messages must be controlled. The main prevalence of using the dispatcher pattern are (i) adjustability (ii) capability to adopt future advancements easily by securing in a new module, (iii) capability to add context in various cumulative steps and (iv) let each functional body be hosted by a third party.

Anyway, this approach may impose overhead and a single point of failure.

3.3 Security View

Various security schemes are possible, based on the capabilities of the promoting system and the possible security services. Related components are situated in the security component class.

3.3.1 Security Component Class

In this class it contains a different component which develops algorithms and techniques for content protection, such as watermarking, thumb impression, secure data enclosing, secure interaction, certificate handling, creation of keys, mutual exchange of keys, management of keys and encryption. These components are organized in layers.

4 ASSESSMENT

This part describes how the presented framework supposes interoperability, alterability and enhancebility [9]. After abstracting the merits, the framework is examined with similar work in DM Project.

4.1 Interoperability

The mediator pattern suggests interoperability because it can instantly transforms messages. Hence, various components can interact with each other. Interoperability is attained by recognizing the main DR Management services as framework components.

4.2 Alterability

Alterability of the system's aspects of particular application rules are supported by the mediator, in which rules can be implanted.

4.3 Enhanceability

Enhanceability is attained by facilitating to include security components by enrolling them at the mediator which clearly maps customer requests and business rules on to the available algorithms[10]. These patterns also organize the association of existing security components and new ones and hence enhance the enhanceability.

4.4 Connection to DM Project

The DM project fully flashes on interoperability which is very valuable thing for DR Management and DM Project.

	Framework style	Merits
Interoperability	Mediater, service components	Reworked service components, message Translation, Transparently Described interfaces
Alterability	Mediater, service components	Attitude described in pluggable strategies
Enhancebility	Mediater, service components	Security components Enrolled at mediator

TABLE I

OVERVIEW OF IMPORTANT PREREQUISITES, THE FRAMEWORK STYLES THAT ENCOURAGES THEM, AND THE ADVANTAGES THEY OFFER.

Framework describes users (e.g, customers, promoters, inventors) as entities that execute preliminary functions which represent the DR Management services that take up digital data. The preliminary functions are connected to the three component classes recognized in the framework proposed in this paper.

Evaluation of the DM Project framework by the given specifications tends to the following outcome [11] [12]. DM Project achieves interoperability with in a single value by allowing preliminary functions with transparently described interfaces various primitive functions from various business person and can be framed into entities that run at the customer, promoter or inventor.

DM Project provides a platform for enhancebility hence latest preliminary functions can easily be implemented [13]. Anyway, the defect of DM Project is lacks a blueprint framework that directs implementers in constructing inconsistent set of functionality, implementation of application is very complex and may impose some hurdles for software bugs.

DM Project some what supports alterability because preliminary functions may have interdependencies, which protects the replacement by other developments. Accordingly, difficult alterations seem to flash on functional considerations, for example by restoring a preliminary function and not on the performance of a system.

In our view, the described DR Management software framework agreeable to an interface particularization as described by DM Project. The latter describes how a individual service be utilized by defining its interfaces and preceded one describes how these are constructed into a complete DR Management system by providing a blue print framework.

5 Conclusion

In this paper, we presented a detailed DR Management framework that provides a report to three main demands in DR Management: accessibility of a common core framework, which can quickly be personalized to differing and dissimilar specifications, and which guides the specific framework drivers (perceive, enhance, customize). The framework consists of three component classes which offer (1) specific services, (2) Illustration and analysis services, and (3) security services. In this paper, we presented the framework from three specific perspectives, beginning from the specific and then enhancing it with the customer tracking, and content securing. By way of valuation, the paper has estimated to what extent three specific requirements are provided by the framework.

With reference to future work, we would like to depict the framework back to contemporary DR Management Systems, and use it to carry out a proof of concept paradigm. Hence, we started research by examining contemporary DR Management systems and framing their utility in a generic software framework, we are sure that it is adequately detailed to visualize the creation and management of DR Management systems and content assessment applications. There is no comparable research proceedings have been presented so far, as best of our knowledge.

References

[1] Automating Production of Cross Media Content for Multi-channel Distribution (AXMEDIS), 2006. http://www.axmedis.org/.

[2] Software framework Glossary, May 2006. http://www.sei.cmu.edu/framework/glossary.html.

[3] L. Bass, P. Clements, and R. Kazman. *Software framewok in Practice*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1998.

[4] E. Becker, W. Buhse, D. Gunnewig,¨ and N. Rump, editors. *Digital Rights Management - Technological, Economic, Legal and Political Aspects*, volume 2770 of *Lecture Notes in Computer Science*. Springer, 2003.

[5] L. Chiariglione. Digital Media Project (DM Project), 2006. http://www.dmpf.org.

[6] P. Gutmann. Cryptlib Security Toolkit, Sept. 2005. ftp://ftp.franken.de/pub/crypt/cryptlib/manual.pdf.

[7] P. A. Jamkhedkar and G. L. Heileman. DR Management as a layered system. In *DR Management '04: Proceedings of the 4th ACM workshop on Digital Rights Management (DR Management 2004)*, pages 11–21, New York, NY, USA, 2004. ACM Press.

[8] P. A. Jamkhedkar and G. L. Heileman. DR Management Interoperability Analysis from the Perspective of a Layered Framework. In *DRM '05: Proceedings of the 5th ACM workshop on Digital Rights Management (DR Management 2005)*, pages 17–26, Alexandria, VA, USA, 2005. ACM Press.

[9] S. Michiels, K. Buyens, K. Verslype, W. Joosen, and B. D. Decker. DRM interoperability and reusability through a generic software. *INDICARE Monitor - About Consumer and User Issues of Digital Rights*, 2(11):16– 20, Jan. 2006.

[10] S. Michiels, K. Verslype, W. Joosen, and B. D. Decker. Towards a Software Architecture for DR Management. In *In Proceedings of 5th ACM Workshop on Digital Rights Management (DR Management 2005)*, Nov. 2005.

[11] W. Rosenblatt, W. Trippe, and S. Mooney. *Digital Rights Management: Business and Technology*, chapter 5. Hungry Minds, Inc., 2001.

[12] IBBT e-paper project, 2006 https://projects.ibbt.be/epaper/.

[13] X. Wang, T. Demartini, B. Wragg, M. Paramasivam, and C. Barlas. The MPEG-21 Rights Expression Language and Rights Data Dictionary. *Multimedia, IEEE Transactions on*, 7(3):408–417, Jun. 2005.