

Flexible Application Rights Management in a Pervasive Environment

Ivana Dusparic, Dominik Dahlem, Jim Dowling
Distributed Systems Group, Department of Computer Science
Trinity College Dublin

E-mail: {ivana.dusparic, dominik.dahlem, jim.dowling}@cs.tcd.ie

Abstract

The development and wider use of wireless networks and mobile devices has led to novel pervasive computing environments which pose new problems for software rights management and enforcement on resource-constrained and occasionally connected devices. Software vendors are, however, still applying old usage rights models to a platform where application rights will be specified, managed and distributed in new and different ways. The characteristics of pervasive environments, such as occasional connectivity, require the introduction of more flexible usage rights models, such as audit-based model, that do not assume the availability of network connections. In this paper we describe a pervasive application rights management architecture for both desktop and mobile applications that provides an integrated platform for the specification, generation, delivery, and management of application usage rights based on Web Services standards. We also introduce flexible usage rights models required by pervasive environments that can be embedded in target applications using aspect-oriented technology.

1. Introduction

The trend towards increasingly pervasive computing environments is brought about mainly by advancements in mobile devices, wireless technologies and the Internet. The movement beyond personal computers to mobile devices introduces challenges in dealing with limited resources, occasional connectivity, and often mobility. Pervasive computing devices connect to global networks or to each other via expensive wireless wide-area networks (e.g. the Internet) or via free short range networks. The latter opens new channels for super-distribution and sharing of software applications that do not impose a cost on the user.

Software applications for pervasive devices are generally designed based on the above characteristics of the pervasive environments. However, the mechanisms for license

management of pervasive applications are still based on the characteristics of fixed networks [4]. The movement of license management into the mobile application space has so far not resulted in the appearance of new license architectures that utilise mobile device features such as super distribution of applications over free communication channels and the separate delivery of application usage rights. Traditional License Management Systems (LMS) are primarily deployed for enterprise desktop applications, and Digital Rights Management (DRM) systems are more concerned with the management of usage rights on content [8], rather than applications. Consequently, there is a lack of a pervasive rights management architecture that supports more flexible usage rights that deal with usage rights on features rather than the full application, and is independent of network connectivity.

This paper presents a Pervasive Application Rights Management Architecture (PARMA) that combines features from both LMS and DRM to provide rights management suitable for pervasive environments. Section 2 of this paper analyses the existing LMS and DRM systems and the problems posed when they are applied to pervasive environments. Section 3 presents the overall PARMA architecture. Section 4 describes the implementation choices that make the architecture suitable for pervasive computing environments. Section 5 evaluates our approach to usage rights management and compares it to existing application usage rights management architectures. Section 6 summarizes the findings and concludes the paper.

2. Background

LMSs are distributed software applications that manage software licenses through their full life-cycle, including the license generation, distribution, enforcement and revocation. Existing systems generally consist of a central licensing server and distributed client libraries linked to licensed applications [4]. Existing LMSs were designed for a LAN environment that assumes constant network connectivity,

and they require a network connection for license enforcement. If the network is not accessible at the time of application start up, application execution will not be allowed [1, 4]. Relying on a network connection for the license enforcement is not feasible in the mobile environment as occurrences of network disconnections are more frequent [9]. Also, frequent network communication with the licensing server adds unnecessary costly network traffic to mobile applications. Another disadvantage of existing LMSs is that they require the addition of calls to the licensing API in the original application source code. These calls are scattered around the application's source code, as license may need to be enforced throughout the application execution. The addition of enforcement source code imposes extra work on the application developer increasing the software development time.

DRM systems are designed to prevent the illegal use of copyrighted information by specifying and enforcing usage rights on that information [11, 12]. Usage rights are specified using a Rights Expression Language (REL)[10]. DRM systems were traditionally concerned with content usage rights enforcement, but are now also specifying and enforcing rights on mobile applications, e.g. OMA DRM [3]. However, the set of usage rights models on content is smaller than a possible set of usage rights models on an application. Usage rights are validated only when a user attempts to execute the application. If the user does not own the right to execute the application, application execution is terminated [3]. This all-or-nothing means of license enforcement does not support fine-grained usage models such as a feature-based model. In such feature-based models, usage rights have to be validated and enforced before every major feature in the application is executed to determine if users have rights on the execution of that feature. Another usage model that existing DRM systems do not implement is the utility-pricing or audit-based usage rights model. This is a form of a post-pay model where actual application usage is logged and users are charged for the application usage afterwards, usually monthly [4]. Market research shows that this is one of the preferred models for application usage billing by customers [15] although it is not provided by existing LMS and DRM systems.

2.1. Motivation

LMSs and DRM systems need to be adapted to the mobility and resource-constrained characteristics of a pervasive environment. At the same time, they should also meet the requirements for traditional fixed network environments in order to provide a complete and integrated platform for license management from a single point within the enterprise. The requirements of an integrated license management systems for pervasive environments are as follows:

- The need for a network connection at the time of license enforcement must be eliminated.
- The amount of network traffic must be minimized to reduce the cost associated with the license delivery and enforcement.
- The amount of additional work by the developer required to license the application must be minimized.
- A variety of usage rights models must be supported, to meet the needs of both end-user customers and enterprise customers in both mobile and desktop environments.

Also, super-distribution of applications must be enabled and encouraged. In this way, applications will reach a larger number of end users, potentially increasing the application popularity and sales, while consumers will benefit from obtaining the application off their peers via free distribution channels, such as Bluetooth and WiFi, rather than via costly wide-area network communication channels.

3. Architecture Overview

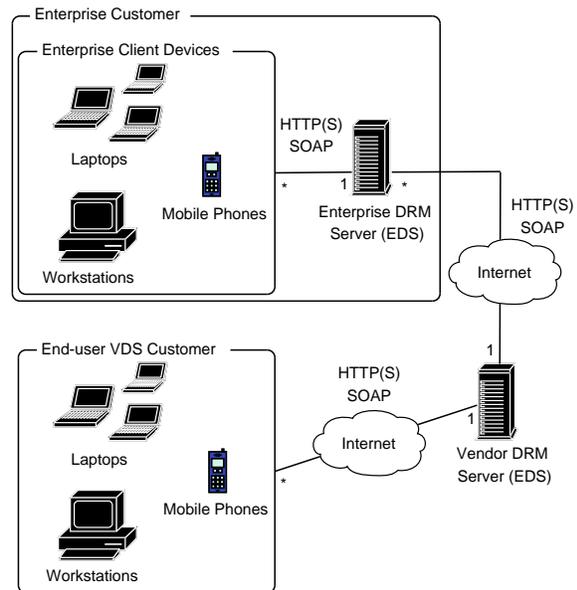


Figure 1. PARMA architecture

The pervasive application rights management architecture described in this paper consists of two servers, a Vendor DRM Server (VDS) and an Enterprise DRM Server (EDS), and client devices running licensed applications (see figure 1). The VDS is designed to manage application usage rights on behalf of one or multiple application vendors. All developers, whether large vendors or small application providers,

can register applications with the VDS and select the associated usage rights models. End-users wishing to download applications and usage rights communicate directly with the VDS, as do enterprise administrators who wish to obtain usage rights for enterprise applications. The EDS server is hosted by enterprises that use one or multiple licensed applications, and it manages applications and licenses on behalf of the enterprise.

The set of features on the EDS is similar to the set of features on the VDS; the main difference is that the EDS deals only with the employees within the licensed enterprise, while the VDS deals with the end-user customers and enterprise customers who wish to use the vendor's applications.

4. Pervasive Rights Management

PARMA defines a new REL, extended from Open Digital Rights Language (ODRL), that can define fine-grained usage models for applications [7]. The PARMA REL can express novel usage rights models: a feature-based model, that is used to limit usage rights on the specific methods within the application, and a post-paid audit-based model, that bills the clients based on the usage data at a later date. The PARMA REL can also define traditional usage rights models such as node-locked[4]. By offering a variety of usage rights models, most of them requiring payments at set intervals rather than once-off and up-front, PARMA is offering flexibility in the usage rights definition required by both vendors and customers [15].

PARMA also provides a novel approach to integrating usage rights in the target application. Usage rights are expressed in XML file that conforms to the PARMA REL. Usage rights files are converted into aspect code, and woven into the original application without any modifications to the original application's source code.

4.1. Usage Rights for Occasionally Connected Devices

The PARMA implementation supports ten different usage rights models, although, the architecture is extensible and new usage rights models can be added. Currently supported models are: unlimited, user-locked, time-limited, feature-based, subscription-based, pay-per-use (real-time or audit-based), node-locked, counted, metered, and concurrent usage rights model. All but two of these models are designed to be fully enforced by the client architecture and as such do not require network connection for the usage rights validation and enforcement. The concurrent usage rights model and audit-based usage rights model require a network connection for validation, but only occasionally, generally

weekly or monthly, depending on the preferences of the application vendor.

The concurrent model in PARMA is implemented similarly to other concurrent licensing models for mobile environments [6], as a time-limited usage rights model with short time-outs, specified in hours or days. The maximum duration for a concurrent model is set to be relatively short, so single devices do not tie up the usage rights for longer than necessary. Usage rights are made available to other devices after the specified time period has expired.

Audit-based pay-per-use model enables utility pricing of the application usage. Variations of post-paid pay-per-use models are implemented for management of usage rights on mobile devices, but they require the network connectivity in order for application to be executed [16]. PARMA implements an audit-based pay-per-use model that does not depend on the presence of network connectivity. PARMA enables the use of an application for a certain period, logging of the usage information, and the payment for the past usage at the end of a defined period. Usage logs are securely stored on the device, and at the end of a set period transferred to the server where the usage information is used to determine payment amounts. Information that can be logged about the application is specified in the usage rights file and is specified by the application vendor. If the information about the particular method needs to be logged, the application vendor needs to provide full signatures of those methods.

4.2. Integrating Usage Rights into Applications

Usage rights using the PARMA REL are expressed in XML documents. XSLT sheets are used to convert XML documents into Java aspect-oriented (AO) code [14]. Aspect-oriented programming (AOP) enables the weaving of REL-specific code into the original application, without modifying the application's source code. Methods in the original application, where usage rights enforcement code should be inserted, are specified when defining the usage rights. Method signatures are converted into pointcut elements in the usage rights file, that are further converted into pointcuts in AO code [14].

Code specified in a Java aspect file is woven in the original application before or after a specified pointcut. For the feature-based usage model these pointcuts represent methods before which usage rights should be validated, and if execution of the given method is not allowed, the enforcement architecture prevents the method from executing. For the audit-based usage model, pointcuts can represent the methods whose execution should be logged as the billing information. In all of the other usage models, pointcuts are inserted at the application start up, and the enforcement architecture either allows the execution to proceed or shuts down

the application, based on the permissions and constraints in the usage rights file. The advantage of using XSLT transformations and AOP techniques to insert the usage rights enforcement code is that the process of adding the usage rights enforcement in the target application is automated and does not require additional programming by the developer.

5. Evaluation

This section evaluates contributions of PARMA to the application usage rights management in pervasive environments. It compares PARMA to existing usage rights management architectures used in pervasive environments, focusing on its suitability for occasionally disconnected devices, the flexibility of usage rights models, and the means of integration of the rights enforcement code with the target application.

5.1. Rights Management in a Pervasive Environment

The use of Web Services technologies to implement the PARMA architecture enables the integration of rights management for applications running on all clients regardless of the programming language the application is developed in, or the platform applications are running on. PARMA registration and application management interfaces are specified using WSDL. PARMA clients can invoke web services over SOAP protocol, or can access the servlets over HTTP(S) protocol, that provide clients to the PARMA interfaces. A rights specification and generation tool is provided as a Java servlet, accessed over the HTTP(S) protocol, so rights can be requested, specified and generated by any client with an HTML browser and HTTP(S) connection. If the device wishing to use the application does not have HTTP(S) support, usage files can be created on a different device and delivered separately from the application over any available communication channel.

Such a design enables PARMA registration and application management interfaces to be called from any device supporting SOAP over HTTP(S), and usage rights files can be interpreted on any device capable of processing XML files. Desktops, smart phones, Personal Digital Assistants, Pocket PCs, and mobile phones with MIDP 2.0 support, all meet the above requirements.

Platform independence and minimal client requirements are a distinct advantage of PARMA when compared to the most widely used existing LMS, FLEXnet. FLEXnet is supported on 26 different platforms, but there is a separate version for each platform, making upgrades and maintenance more difficult. Supported platforms do not include any of the operating systems used on mobile devices, so FLEXnet

is suitable only for desktop environments rather than pervasive ones [4].

The majority of PARMA usage models do not require an Internet connection for usage rights management process. Eliminating the need for constant network connection is important in the pervasive environment to support application execution during periods of network disconnection, and to minimize costs associated with network traffic initiated by the usage rights enforcement architecture. Both application and usage rights can be transferred to a device from another device over short range protocols, so an expensive Internet connection is not required at any stage of the usage rights management process.

Support for the disconnected application execution on the mobile devices gives PARMA an advantage when compared to other usage rights management systems. FLEXnet does not require a network connection for any model apart from the concurrent model but is applicable only in the fixed network environment where network connectivity is not an issue. XSLM, the only existing proposed open standard for the integrated license management, requires that a network connection is available for the duration of the application execution [1].

Some licensing models for consumer mobile applications, such as models supported by OMA DRM, do not require an Internet connection for rights enforcement. However, flexible usage models such as pay-per-play are supported only when devices are connected to the Internet. PARMA supports an audited pay-per-play model (audit-based model) that gathers usage data on the device, without requiring a network connection, and transfers it to the server in time intervals set by vendor, once a connection becomes available.

5.2. Flexible Usage Rights Models

The PARMA project defined a new REL, PARMA REL, extended from ODRL to support a variety of usage rights models for both desktop and mobile applications. Other existing RELs (ODRL, OMA DRM, XrML) are developed either for desktop applications or content, and as such do not offer the expressiveness necessary to meet PARMA requirements. PARMA supports all of the traditional usage rights models applied to desktops, but also introduces feature-based and audit-based models. PARMA REL is designed to be open, extensible and compatible with current standards.

As shown in table 1, PARMA REL supports the widest range of usage rights models for software applications, when compared to other existing RELs. As such, PARMA is more suitable than these RELs for the expression of usage rights on applications, primarily on mobile devices, but also desktops.

	OMA 1.0	OMA 2.0	ODRL	XrML	PARMA
Node		x		x	x
Concur	x	x	x		x
PrePay			x	x	x
PostPay			x	x	x
Audit					x
Time	x	x	x	x	x
Metered		x	x	x	x
User			x	x	x
Feature					x
Subscr.	x	x	x	x	x

Table 1. PARMA vs other RELs [2, 3, 5, 13]

LMS FLEXnet supports all of the usage rights models that PARMA supports but these models are supported only on desktop devices and laptops. FLEXnet does not use a REL to express usage rights; the usage rights are expressed in license files using a proprietary format. PARMA expresses usage rights using an open REL that is compatible with standardized ODRL and OMA DRM usage rights architectures, rather than using proprietary formats.

5.3. AOP for Usage Rights Management

Existing LMS require addition of the calls to enforcement API to the application's source code, while all of the PARMA's enforcement code is specified in a separate aspect class.

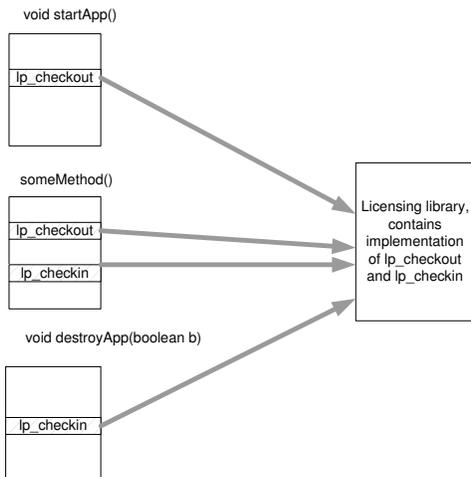


Figure 2. Cross-Cutting Usage Rights Enforcement (FLEXnet)

Figure 2 shows the feature-based model, implemented using the FLEXnet license management systems, that performs the license validation at a MIDP application start-up,

method *startApp*, application shut down, method *destroyApp*, and prior to the execution of method *someMethod*. Source code of all of the three methods needs to be modified in order to insert the rights enforcement code.

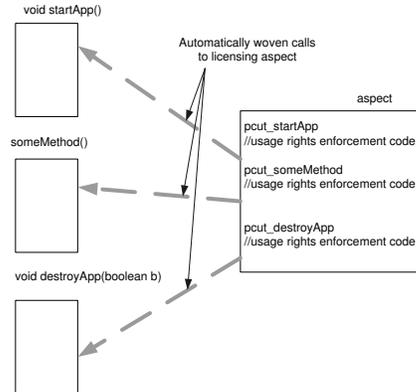


Figure 3. Usage Rights Enforcement Encapsulated in an Aspect (PARMA)

Figure 3 shows the feature-based model that performs the license validation at the application start up, method *startApp*, application shut down, method *destroyApp*, and prior to the execution of method *someMethod*, using PARMA. The original methods are not modified, as all of the enforcement code is specified in a separate class.

Aspects are generated from usage rights files, so insertion of rights enforcement code does not require knowledge of proprietary APIs or any programming knowledge at all. The only information application providers needs to supply are full signatures of methods that usage rights are specified on. In the case that usage rights are specified on the full application, the only information that needs to be provided is the entry point to the application.

The rights enforcement aspect can be woven into target application even if the original source code is not available, but is woven into compiled classes or packaged jar file. Encapsulating the rights enforcement source code into a single aspect file reduces the time and amount of work needed in order to enable the application for distribution using a certain usage rights model. Rights enforcement can be added and removed from the application by simply recompiling the classes with or without the aspect containing the rights enforcement code. Also, in order to be enabled for a distribution with a different usage rights model, the original application only needs to be recompiled with a different aspect file implementing the new usage rights model, rather than modified. Separating the application source code from the rights enforcement code improves the source code modularity, simplifying maintenance of both target application

and rights enforcement code.

6. Conclusions and Future Work

Existing LMS and DRM systems are suitable for the desktop and LAN environments they were originally developed for, but have several pitfalls when applied to the pervasive environment. DRM systems have a primary focus on content rather than applications, and as such they lack support for flexible fine-grained usage models for applications. Existing LMS are primarily designed for the desktop environment and make an assumption of constant network connectivity, while devices in pervasive environment have only occasional connectivity. Also, LMS require developers to modify the source code of the application by inserting the calls to the licensing API, and consequently increasing the time required for application development.

PARMA took several novel approaches to usage rights management and combined features from existing LMS and DRM systems, to provide an integrated platform for management of application usage rights in pervasive environments. PARMA supports a variety of usage rights models, including models specifically tailored to occasionally connected devices, such as audit-based, and fine-grained control of application execution, such as feature-based. Usage rights enforcement is integrated into applications using usage rights files converted into AOP code and woven into the original application, removing the need for including the enforcement source code manually by the application developer. PARMA is suitable for usage by variety of client devices, as the only criteria that devices need to satisfy in order to use PARMA is HTTP(S) and SOAP support. These three characteristics make PARMA suitable for application management in pervasive environments.

The current version of PARMA supports the automatic generation of MIDP 2.0 source code that can run on mobile phones. Future work on PARMA will involve implementing the additional XSLT transformation sheets to support the transformation of usage rights into other programming languages and rights management of applications on other platforms. The footprint of the SOAP clients interacting with the licensing server will be minimized so clients can run directly on resource constrained devices, rather than communicating with SOAP clients via servlets, as it is the case in current PARMA implementation. The full client side enforcement architecture will be added that interprets the usage rights files, and according to the provider's preferences and settings, deals with invalid usage rights. Finally, as PARMA is based on web services, it can also be integrated with back-end services, such as payment, to provide an integrated usage rights management and payment solution for individual and enterprise customers.

References

- [1] Systems Management: Software License Use Management (XSLM). Technical standard, The Open Group, March 1999.
- [2] ISO Multimedia Description Schemes Group. text of ISO/IEC CD 21000-Part 6 - Rights Data Dictionary (RDD), July 2002. http://www.chiariglione.org/mpeg/working_documents.htm.
- [3] Open Mobile Alliance. Digital Rights Management Version 1.0, September 2002.
- [4] Flexnet, 2004. <http://www.macrovision.com/products/flexnet/index.shtml>.
- [5] Open Mobile Alliance DRM Rights Expression Language Version 2.0, April 2004.
- [6] XTNDConnect Server GE Frequently Asked Questions, August 2004. <http://resolution.extendedsystems.com/esi/service+support/xtndconnect+server/faqs/default.htm>.
- [7] D. Dahlem, I. Dusparic, and J. Dowling. A Pervasive Application Rights Management Architecture (PARMA) based on ODRL. In R. Iannella and S. Guth, editors, *Proceedings of the First International Workshop on ODRL*, April 2004.
- [8] M. Fetscherin and M. Schmid. Comparing the usage of digital rights management systems in the music, film, and print industry. In *Proceedings of the 5th international conference on Electronic commerce*, pages 316–325. ACM Press, 2003.
- [9] R. Gray, D. Kotz, S. Nog, D. Rus, and G. Cybenko. Mobile agents for mobile computing. Technical report, Dartmouth College, 1996.
- [10] S. Guth. *Lecture Notes in Computer Science*, chapter Rights Expression Languages. Springer-Verlag Heidelberg, 2003.
- [11] S. O. Hwang, K. S. Yoon, and C. S. Park. Design and Implementation of a Licensing architecture for Distribution of Copyright-Protected Digital Contents. *Telecommunications Review*, 12(5), October 2002.
- [12] R. Iannella. Digital Rights Management (DRM) Architectures. *D-Lib Magazine*, 7(6), June 2001. <http://www.dlib.org/dlib/june01/iannella/06iannella.html>.
- [13] R. Iannella. ODRL Specification 1.1, August 2002. <http://odrl.net/1.1/ODRL-11.pdf>.
- [14] I. Kiselev. *Aspect-Oriented Programming with AspectJ*. SAMS, 2002.
- [15] A. M. Konary, S. Graham, and L. A. Seymour. The Future of Software Licensing: Software Licensing Under Siege. Technical report, IDC, March 2004.
- [16] ninemsn. Australia's First Real-time Multi-player Mobile Game Launched, August 2004. <http://games.ninemsn.com.au/article.aspx?id=14624>.