The Technology of Rights: Digital Rights Management

By Karen Coyle

Based on a talk originally given at the Library of Congress, November 19, 2003

Introduction

You have probably heard the expressions "thin copyright" and "thick copyright" referring to different philosophies about copyright law. Very briefly, thin copyright usually refers to a minimalist approach to copyright, giving works only as much protection as is needed to encourage creativity but with a goal of making works readily available to the public. Thick copyright is a more maximalist approach, and crudely put the goal of thick copyright is generally to maximize profits. We appear to be moving toward thick copyright, not only in this county but around the world in general. This movement is being spearheaded, as you might expect, by companies whose main product is in the form of intellectual property, such as books, movies and music.

But there is yet another trend relating to the protection of intellectual property and that is the creation of technological controls to protect digital works. This is referred to as Digital Rights Management, or DRM. DRM is not a single technology and it is not even a single philosophy. It refers to a broad range of technologies and standards, many of which are still in the planning and development stage. DRM is not thin copyright, and it isn't even thick copyright; DRM is potentially a nearly absolute protection of works.

> "With the development of trusted system technology and usage rights languages with which to encode the rights associated with copyrighted material, authors and publishers can have more, not less, control over their work."
>
> Mark Stefik, Shifting the Possible: How Trusted Systems and Digital Property Rights Challenge Us to Rethink Digital Publishing
>
> Berkeley Technology Law Journal, v. 12, n. 1, Spring, 1997 [1]

In the remainder of this article we will look at how DRM protects works today and where this technology may be headed in the future.

Why Do We Need DRM?

What's the motivation behind digital rights management? Let me give you a simple illustration. Say I have a book, a hard copy book, that I have borrowed from the library. Maybe I would like to have a copy of my own. What prevents me from going down the street to a copy center and making myself a copy of the book? Well, it's copyright law that prevents me from doing that, isn't it?

In fact, that's not true at all. Copyright law does not **prevent** me from making the copy. It may make me feel a bit guilty about making the copy, or I might fear getting caught, but it doesn't prevent me from making the copy. Yet I am unlikely to copy the book. Why is that? Because I don't want to spend an hour and a half at a copy center opening the pages and punching the copy button. Because in the end the copy will cost me as much or more than buying a copy of the book in paperback. And because what I will end up with is a poor copy on bad paper in an 8.5x11 format, unbound. In the end, making a copy of a hard copy books is uneconomical, in terms of time and money, and the result is pretty undesirable.

Now let's say that I have the very same book in a digital format. If I want to make a copy, I can make that copy almost instantly. It will cost me nothing. And the end result will be a perfect copy of the original. Not only that, I can make one hundred or a thousand copies almost as easily as I can make one. I can email the file to everyone in my address book, or I can place the file on a peer-to-peer network and let anyone on the Internet have access to it. With the digital file, the economics are slanted very much toward making copies.

Note that the digital file is protected by the very same copyright law that the hard copy is, the one that does't really prevent us from making copies. What we have here is the Napster effect, which is based on the ease of copying. And because law doesn't seem to have worked as a preventive measure, there is some justification that only a technology-based protection will ever work to protect digital works.

Why Encryption Isn't Enough

Whenever you talk about protecting digital files, encryption is part of the answer. But the protection that encryption offers has its own limitations. To begin with, encryption does not prevent you from copying a file. Copying a file is just a matter of moving a series of ones and zeroes from one digital device to another, and you can copy an encrypted file just as easily as you can copy an unencrypted one. You can also email it, or place it on a peer-to-peer network. The protection that encryption provides has nothing to do with copying; instead, encryption prevents **access** to the content of a file. And this is a key point to understand about both encryption and about digital rights management: that the controls are on access and on use rather than copying. You can copy an encrypted file dozens of times, but none of those copies have any value if you cannot access the content. And there's very little value in sending such a file to anyone else, unless they have the key.

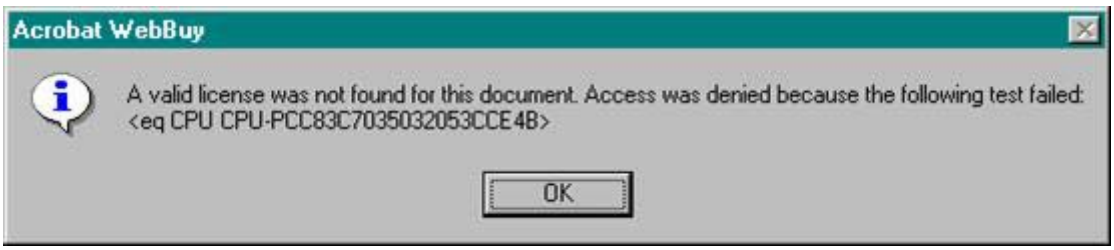
So how does encryption actually help protect files? Let's step through a few scenarios that will illustrate the way that encryption can be used in a DRM system. I'll pretend that I have written an ebook and Jane wants to buy it.

In this first scenario, I encrypt the ebook and send it to Jane, with an email that tells her the key to use to open the file. Have I protected the book? No, not at all. Jane now has the book and she knows the key. She can send the book to all of her friends and tell them the key, so everyone can have a usable copy. Encryption only works when the person holding the key is the one who wants to protect the digital file. Giving the key to anyone else negates the purpose of the encryption. You could argue that I could decide to trust Jane to keep the key to herself, but in general we can assume that Napster, Kazaa, and other trading networks have pretty much eliminated "trust me" as an option that owners of intellectual property will go for.

How can I get the key to Jane without actually giving her the key? One solution is that I can give the key to Jane's computer, not to Jane. In this scenario, Jane buys my ebook and I allow her to download it to her computer. At the same time, she downloads a small file that is also encrypted but that contains the key that opens the ebook. The ebook software that she is using can un-encrypt this key file, often called a "voucher," and can then use the key to open the ebook. Jane never sees the key. And she may be unaware that there is a key file because it may be sent to her machine as a hidden file, or it may be otherwise disguised with an odd name or place on the hard drive.

Does this protect the ebook? No, because if Jane is clever, she can figure out that by making copies of both files and sending them to a friend, that friend can also access the file and read the ebook because she has both files on her computer. Anyone possessing the two files can read the ebook, whether or not they paid for it.

So now our question is, how can we give the key to Jane's computer in a way that Jane can't send it on to others? We do that by tying the key to the identity of Jane's hardware. In this scenario, Jane pays for the ebook. In the exchange that takes place as Jane negotiates her payment between her computer and mine, a program returns to my site some piece of identifying information about Jane's computer. This may be a identification number of her CPU, a serial number from her hard drive or her BIOS. The main thing is that is is something that uniquely identifies Jane's computer and it is not something that she can readily change. Now when Jane opens the file on her machine, the voucher file contains a record of that unique hardware identification, and the program that opens the file will not work if the hardware of the current machine doesn't match the hardware ID in the voucher. If the digital file and the voucher are moved to another machine, the program will not open the file. Instead, the user may see an error message like this one:



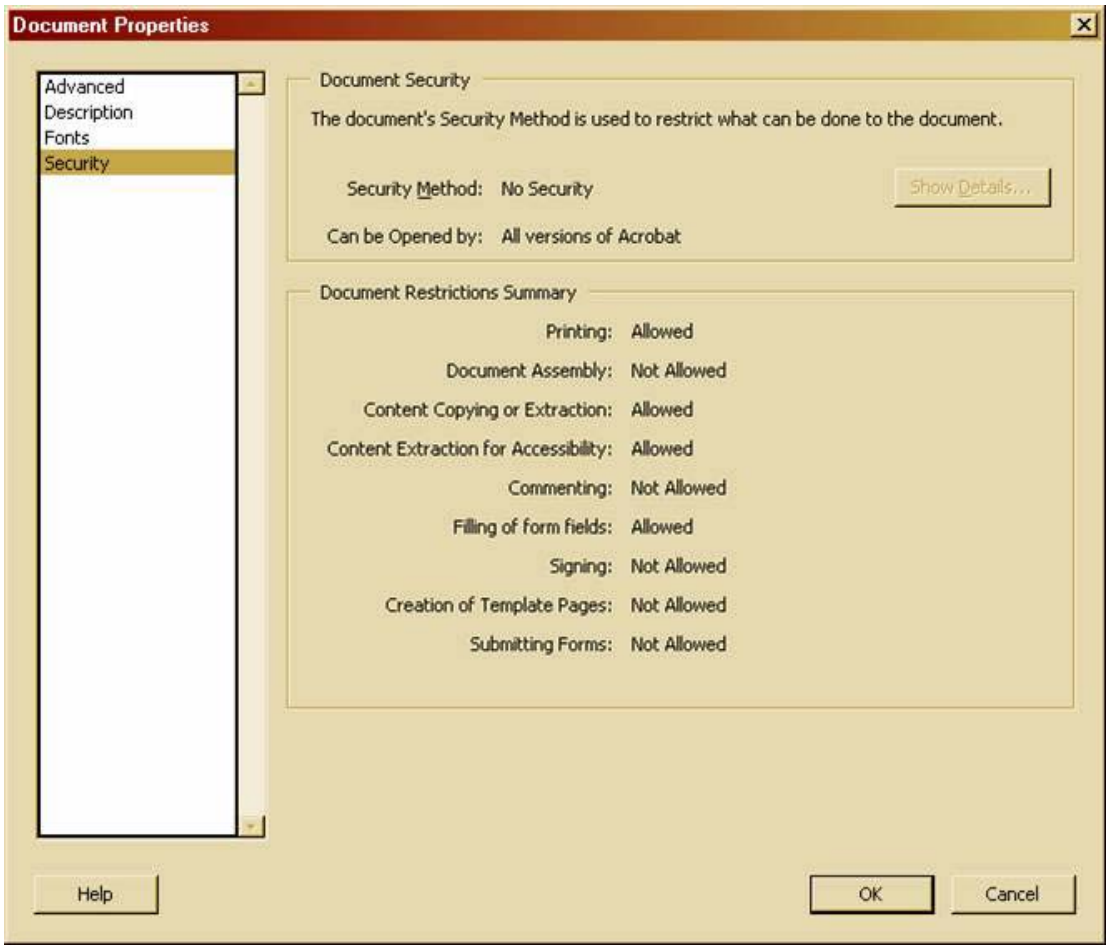
This technique of tying a digital file to a particular piece of hardware is a common DRM solution today. It has obvious problems in a world where the average life of hardware is two to three years, but at the moment it is the best method we have to control access to a digital file. To create a better solution, one that would connect the digital file to a person rather than to a machine. This would allow a person to move files from one computer to another in the way that you pack up your books and move them from one house to another, requires a more sophisticated technology called "trusted systems." We'll discuss the work being done around trusted systems further on in this article, but suffice it to say at this point that trusted systems may be the next level of development in digital rights management.

Usage Rights

You might think that once you have the file open you have surmounted the DRM controls, but in fact the controls over access to the file is only the first step in DRM. Once the file is open there are many more controls that can be applied, such as controls over whether you can print from the file, copy passages to the clipboard, or whether the file expires after some time period. If you read ebooks, you have probably made use of a program with these controls. We'll look at two examples of file protection: the Adobe Acrobat Reader, which can read protected or unprotected PDF files and is a commonly used format for ebooks, and the Microsoft Reader, an ebook reader that uses its own format and protection technology.

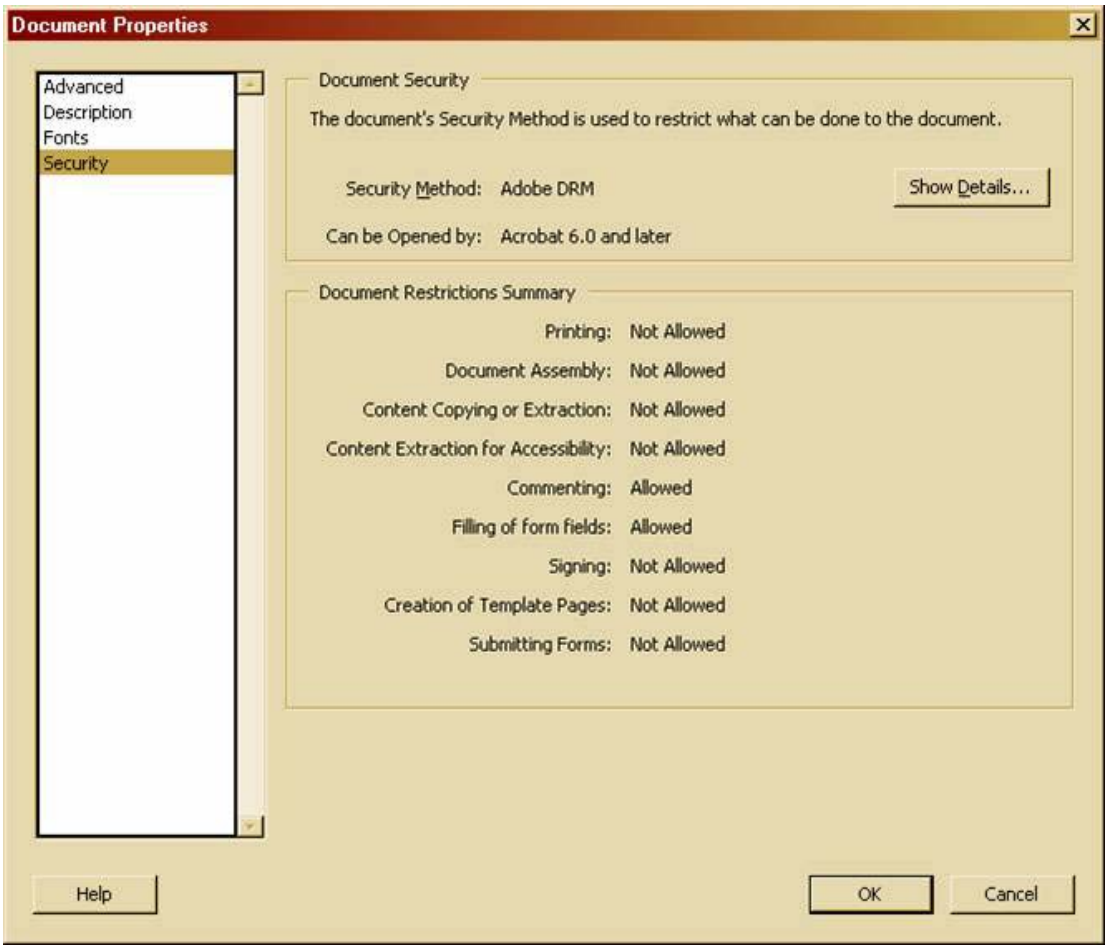
Adobe Acrobat

Adobe Acrobat [2] has had digital protection since the 5.05 version of the Adobe Acrobat Ebook Reader. The 6.0 version of the Adobe Reader combines the capabilities of the Acrobat PDF reader and the ebook reader in a single software implementation. By opening a file and clicking on File/Document properties you can see the protections that exist for that file. In the screen shot below we see the rights statement for an unprotected PDF file opening with Acrobat version 6.0.



As you can see, it states that this file can be opened with any version of Acrobat and that it has no security applied to it. It states that printing is allowed, and "content copying" is allowed, meaning that you can highlight areas of the document to copy to the clipboard function of the computer. Even though the document is not protected, not all of the possible rights are allowed, possibly because of software limitations. For example, it isn't possible in this software to edit the document or to use it to add comments to the file. One important right that is granted is for accessibility for readers with special needs. This accessibility grant may mean that the content of the file can be read out loud by text to speech programs, or can be captured and rendered by programs that increase the size of the font on the screen.

Now we look at the same rights display for a file that does have protection:



Notice that some things have changed here. To begin with, it says that the security method is Adobe DRM. It also says that the file can only be opened in Acrobat 6.0 and later. Presumably that is because Acrobat 6.0 is able to provide the protection necessary. It also says that printing is not allowed. It's not just not allowed, it isn't possible. When you open the file in the Acrobat software there will be no print button, and no print command in the pull down menu. Content copying is also not allowed. This means that the software will not allow you to capture any part of the content to the clipboard; it has essentially disabled the clipboard function for this file. And content extraction for accessibility is also on the "not allowed" list, so presumably the software is able to block the use of text-to-speech software.

Microsoft Reader

Microsoft Reader [3] is ebook reading software, and it is used to open files with the ".lit" extension. Like the Adobe Acrobat software, it has some DRM capabilities. Unlike Acrobat, you cannot easily see a list of the protection properties pertaining to an individual book, but there is a description of these functions in the Reader's help file.

Microsoft reader has three categories of works that correspond to levels of protection. These are described in the help file as:

- Sealed eBooks - Sealed books allow you to use all features of Microsoft Reader, while ensuring the authenticity of content, which means that you cannot modify the text and other content.
- Inscribed eBooks - Inscribed content displays information from the purchaser (for example, the name of the purchaser) on the cover page. This reinforces honest usage by consumers.
- Owner Exclusive eBooks - Publishers provide these eBooks with inscribed security plus an additional encrypted license that requires you to activate your computer for Microsoft Reader before purchasing and reading these eBooks. Publishers prohibit you from copying and pasting text from Owner Exclusive eBooks into other applications. They also prohibit you from using the Text-to-Speech functionality

The Sealed eBooks are only protected from tampering. This is a useful function because it guarantees that the file that you have received has not been modified since it was created. It also prevents you from modifying it, such as replacing the author's name with your own.

The inscribed eBooks have something like a digital book plate saying who is the owner of that copy. By linking the file to someone's name, it is assumed that the person will be more reluctant to distribute copies to friends or over the Internet since it can presumably be traced back to you. (The Palm Reader uses a similar kind of deterrent: the ebook file in the Palm Reader contains the name on the credit card that was used to purchase the ebook. To open the ebook the first time on your device, you must input the credit card number that was used to purchase the book. Both the name and the credit card number have been included in the file that was downloaded. Even if you don't mind having a file with your name on it circulating around the digital space, you are highly unlikely to provide your credit card number so that others can activate the book on their handheld devices.)

It is the Owner Exclusive eBooks that have true DRM protection. These are generally commercially published books that are sold through online bookstores. To open an Owner Exclusive book in Microsoft Reader you must first activate your computer. What does this mean? The Reader help file explains:

> "Activation associates your Microsoft Passport account with the specific copy of Microsoft Reader on your computer. The activation process downloads a software module to your computer’s memory unique to you and your computer. This software module also uses your Microsoft Passport account number and other information unique to your computer, to protect eBook titles against unauthorized copying."

As described in the section above on basic DRM technology, activation essentially creates a unique identification for your computer. The "other information unique to your computer" refers to a hardware identification of some kind. So the books that you purchase cannot be opened on someone else's computer, thus rendering any copying of this file pretty useless.

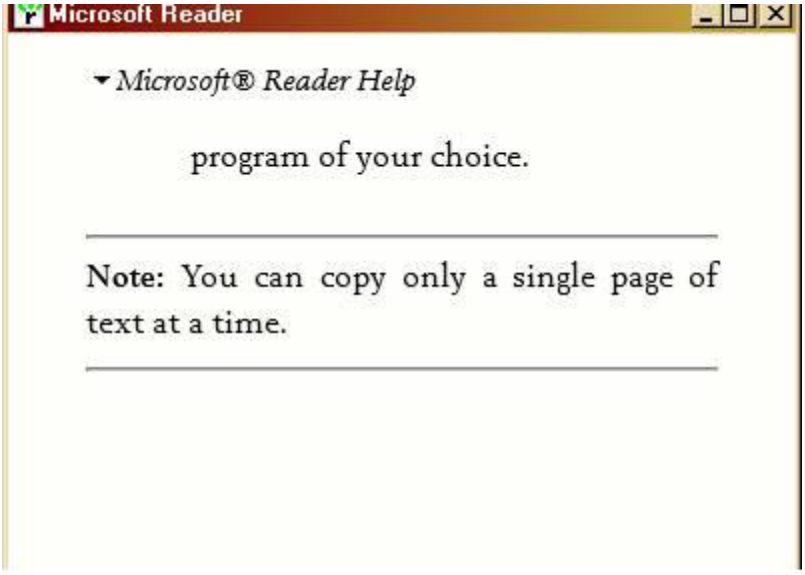
In addition to having this "anti-piracy" function, the Owner Exclusive book also has use restrictions that apply to the legitimate owner of the ebook. At this level of protection the

ebook will not allow content to be copied to the clipboard, so no copy and paste from the text is possible. In addition, the text-to-speech capability that is built into Microsoft Reader for accessibility purposes is disabled. This latter appears to be at the request of publishers whose contracts do not include audio performance rights. It is also rumored that some technologists fear that the text-to-speech could be fed back into a voice recognition program, thus reproducing the written text.

When I search the Microsoft Reader help document for the word "print," this is what I get:

The Microsoft Reader has no print button, no print command in its pull down menus, and no mention of print in its help file. This is true for all documents, not just those that are considered to be protected by DRM. So if you were to convert a public domain document, like a federal government publication that has no restrictions on use, into a Microsoft Reader document, you would not be able to print from it. This is fine as long as we have document readers that are compatible with the public domain, but the desire for secure reading could mean that public domain capabilities are not included in any available reader software in the future.

Even in a file with no digital rights management capabilities there are limitations, deliberate or not, that are not unlike the inconveniences of copying hard copy books. For example, the Microsoft Reader help explains that even where copying is permitted, you can only copy one page at a time. This is because there is only one page at a time presented on the screen so that's all that you can highlight at a given moment. In a book that is not Owner Exclusive you could in theory copy the entire book, one page at a time. The tedium of this would prevent many of us from doing so, and like a photocopy of a paper book the end result would be inferior because the clipboard captures the text from the ebook but not the formatting.



Rights Expression Language

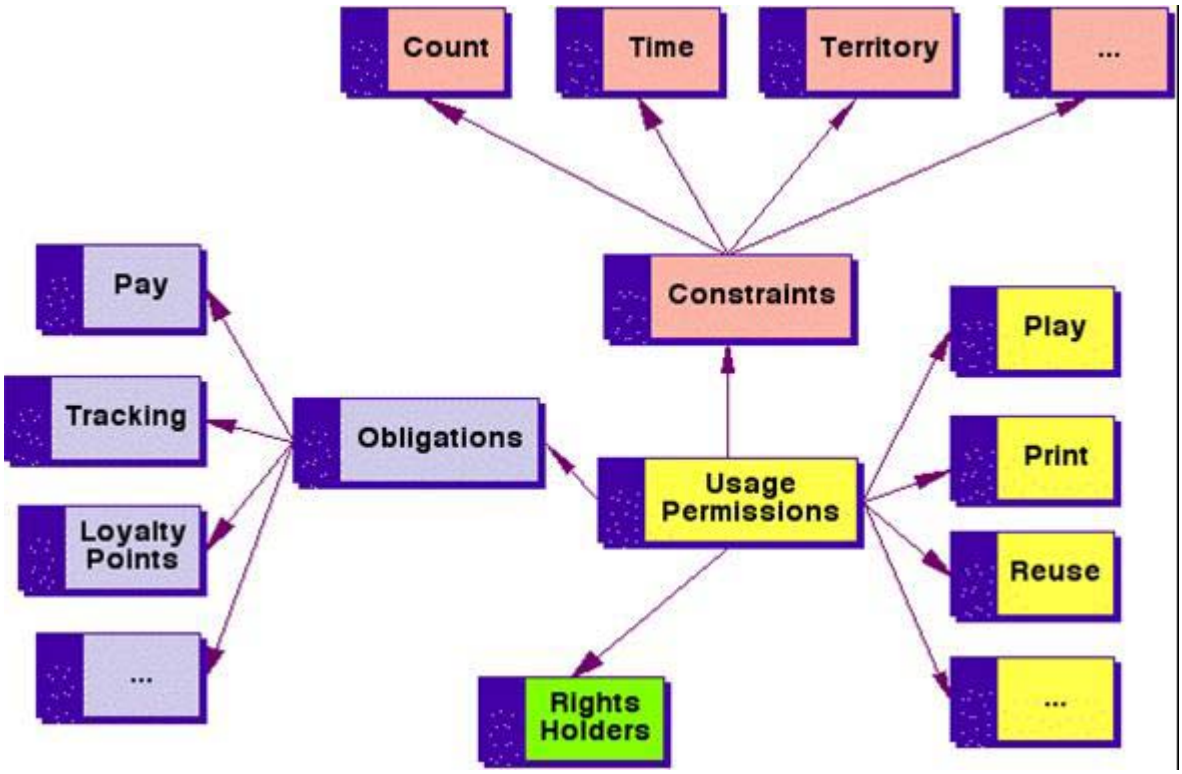
The controls in Adobe Acrobat and Microsoft Reader (as well as those in other protected formats like Palm Reader, MobiPocket, etc.) are fairly limited in scope. They limit access to the file, and those that run on a general-purpose computer also control interaction with a few basic computer functions like printing and copying to the clipboard. But the future of digital rights management is in the development of something called a Rights Expression Language (REL) that will allow a publisher to designate a much more detailed and complex set of usage controls. The REL will be able to control the number of times a text can be read, can set timed controls (i.e., "for two weeks starting today"), and can manage complex relationships of distribution, sale and lending.

So what's a Rights Expression Language? It seems obvious, but it's a language that expresses rights, that says what rights you have in relation to this file. You could argue that we have a perfectly good rights expression language in our copyright laws. After all, the purpose of the copyright law is to state what are the rights of copyright holders and what are the rights of the public. The problem is that the copyright laws are written in legal language, a language that requires some interpretation and some judgment as it is put into practice. An REL in the digital rights management sense is a different kind of language; it is a formal language like mathematics or like programming code; it is language that can be executed as an algorithm. A rights expression language is not open to interpretation but must be rendered precisely through software.

What can a rights expression language control? It's important to keep in mind that a rights expression language must work within the computing environment and the capabilities of that environment and computers are entirely quantitative by their nature. So REL's tend to be about things that a computer can count, like:

- Time. Computers have clocks built into them, and they know what time it is. They know how much time passes between moment A and moment B. They also know what day it is, so they can work with "two weeks" or "every Tuesday." Combine these and they can be made to act on a definition like "from 2:00 p.m. to 2:55 p.m. every other Tuesday in March."
- Units. If you can define a unit to a computer in a way that it understands, the computer can use that in its activities. A unit in an ebook could be a chapter, page, paragraph, sentence or even a word. In a digital film a unit could be a track defined in a DVD file. Using the unit "page" an REL could allow the printing of 5 pages from a book, but no more. Or music could be licensed by the hour or minute.
- Value exchange. In other words, money. An REL fits well into the general ecommerce environment. Prices can be associated with any of the other units of measure, and there are ecommerce systems already that an sell an individual chapter of a textbook for a price that is much less than the cost of the entire book. A price could be associated with a half hour of music listening, or there could be a per-page copying cost. The exchange could also take a non-monetary form, like granting a right to read a chapter to a user who first views a particular number of advertisements. Or allowing people to purchase content in exchange for frequent flyer miles. Any value that can be exchanged could be used.

This diagram, below, by Renato Ianella [4], shows the general overview of a rights expression language, with a triangle of rights, which he calls "usage permissions" (in yellow), constraints like time, units or territory (in pink), and payments or exchange of value (in blue). This triumvirate of rights, constraints and payments is the basis of a general purpose rights expression language, but the simplicity of this model can be deceptive. In fact, a true general-purpose REL is far from simple.



There are some very sophisticated RELs in development although none has been fully realized as a product yet. These languages are very complex because they need to be able to express all of the possible uses of a text or a recording or a film that can be granted to a member of the public that accesses it. They also need to include language to manage business relationships like wholesale and retail distribution.

Open Digital Rights Language (ODRL)

Renato Ianella, whose diagram we viewed above, has developed a rights language that he proposes as an open standard, called Open Digital Rights Language [5], or ODRL. ODRL is still in development but it already has quite a large vocabulary related to usage permissions, constraints and payments. Here are some of the terms it uses:

Rights	Constraints	Payments
excerpt execute give install lease lend modify move play print printer remark restore sell transferPerm uninstall	fixedamount hardware interval network percentage peruse range screen uid version unit	feeType payment postpay prepay transaction

ODRL is rendered in XML. Here's a small fragment of a sample ODRL file. This fragment gives the user permission to display a set number of pages. You can see that the permission is "display," the unit of measure is "NumberOfPages," and the range of pages is a minimum of one to a maximum of five.

```
<o-ex:permission>
        <o-dd:display>
                <o-ex:constraint>
                <o-dd:unit o-ex:type="onix:NumberOfPages">
                <o-ex:constraint>
                        <o-dd:range>
                        <o-dd:min>1</o-dd:min>
                        <o-dd:max>5&ltp;/o-dd:max>
                        </o-dd:range>
                </o-ex:constraint>
                </o-dd:unit>
                </o-ex:constraint>
        </o-dd:display>
</o-ex:permission>
```

eXtensible rights Markup Language (XrML)

XrML [6] is a product of the company ContentGuard, and is based on the early REL work of Mark Stefik, whose quote was shown early in this article. As a language, XrML is more abstract than ODRL and could properly be thought of as a meta-language: a language for developing rights expression languages. Because of this its vocabulary is less immediately understandable than that of ODRL but can be used as building blocks to express specific rights. For example, here are a few XrML data elements:

Grant/forAll
Grant/principal
Grant/right
Grant/resource
Grant/condition
Grant/delegationControl
Grant/encryptedGrant
Principal
The AllPrincipals Principal
The KeyHolder Principal
Right
The Issue Right
The Revoke Right
The PossessProperty Right
The Obtain Right

Although these data elements appear to be very abstract, in fact XrML expresses an REL with great precision. Here is one paragraph from the XrML-based standard that was developed for the Motion Picture Experts Group (MPEG):

> 5.5.2.1 Authorization of Located Bits
> Let *g* be any authorized `Grant` containing a `Resource` *d* which is a `DigitalResource`. Let *b* be the sequence of bits which is the result of any execution of the location algorithm of *d*. Then the Grant *g'* which is identical to *g* except that *d* is replaced by a `DigitalResource` which contains a child binary element which contains a base64 encoding of *b* is also authorized.

The key thing here is that the XrML "language" is not semantic in the sense of a human language, but is expressed as formulas with mathematical-like precision. The resulting rights statement, however, is no more or less comprehensible than the XML that is derived from ODRL:

```
<datum>
  <keyHolder licensePartIdRef="Alice"/>
</datum>
<datum>
  <mx:diReference licensePartIdRef="eBook"/>
</datum>
<datum>
  <oebx:recurrancePeriod>
    <oebx:period>
      <oebx:unit>oebx:day</oebx:unit>
```

```
      <oebx:unitCount>7</oebx:unitCount>
      </oebx:period>
    </oebx:recurrancePeriod>
</datum>
<datum>
    <oebx:exerciseLimitPerPeriod>5</oebx:exerciseLimitPerPeriod>
</datum>
```

This statement is similar to the one we saw for ODRL. It is a fragment of a larger expression, but is giving Alice, the license holder, a license to exercise a right (not named in this fragment) that lasts for a period of seven days.

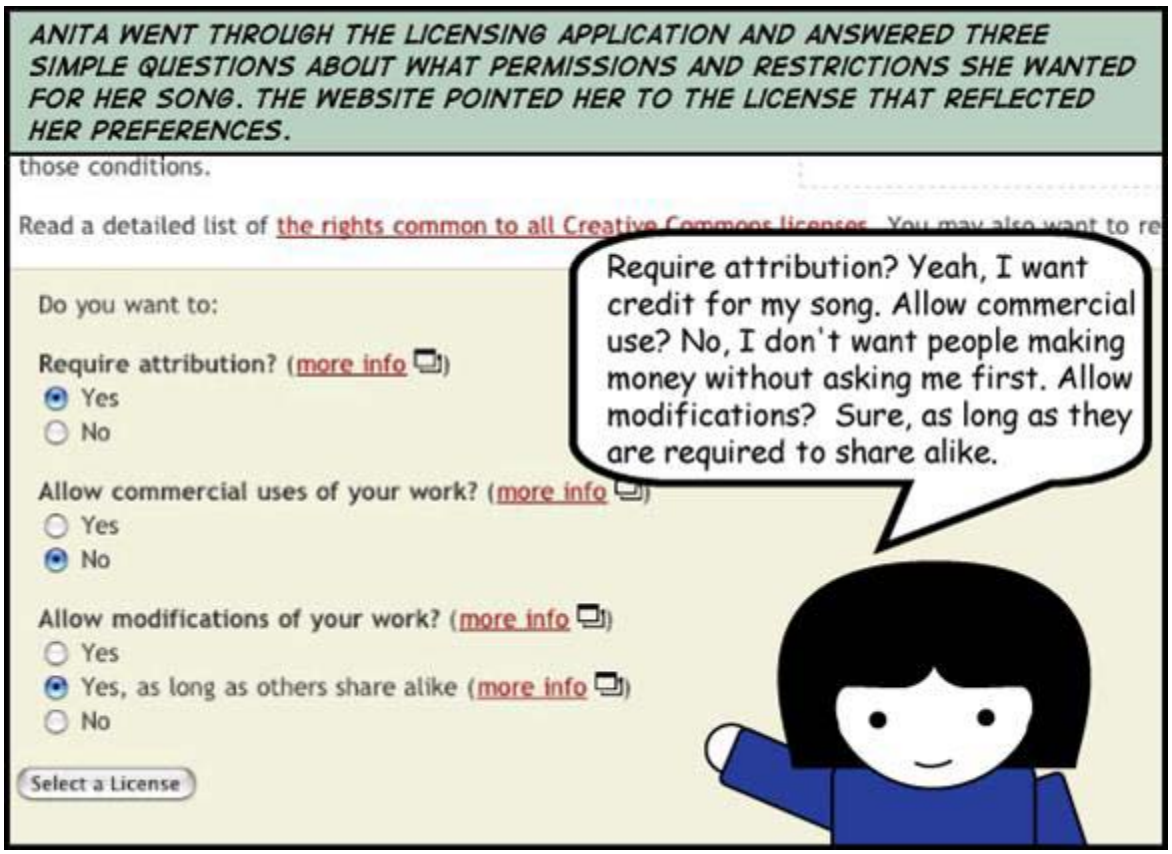
XrML is the basis for the REL developed for Moving Picture Experts Group (MPEG), the group developing standards for digital audio and video. The MPEG standard is an international standard and the rights expression language is incorporated as part 5 of MPEG-21 [7]. This makes XrML the key REL for digital media at this moment in time.

Specialized RELs

ODRL and XrML are examples of generalized rights expression languages, and part of their complexity is that they must cover every imaginable right and constraint. It is possible to develop a rights expression language that is specific to a particular task, and such a language can be more limited in scope. In a sense we have already seen that with the rights that are incorporated in the ebook readers developed by Adobe and Microsoft. Another example is PRISM [8].

PRISM stands for the Publishing Requirements for Industry Standard Metadata. This is an industry standard for syndicated magazine and newspaper articles. PRISM operates in a business-to-business context where the businesses have pre-existing relationships and business contracts. The rights expression portion of the PRISM metadata provides additional information, such as copyright statements and expiration dates on the files. Because the businesses concerned control their relationships through contracts, PRISM is not expected to be utilized in an automated enforcement system. This reduces the need for the kind of formal precision that is required of XrML.

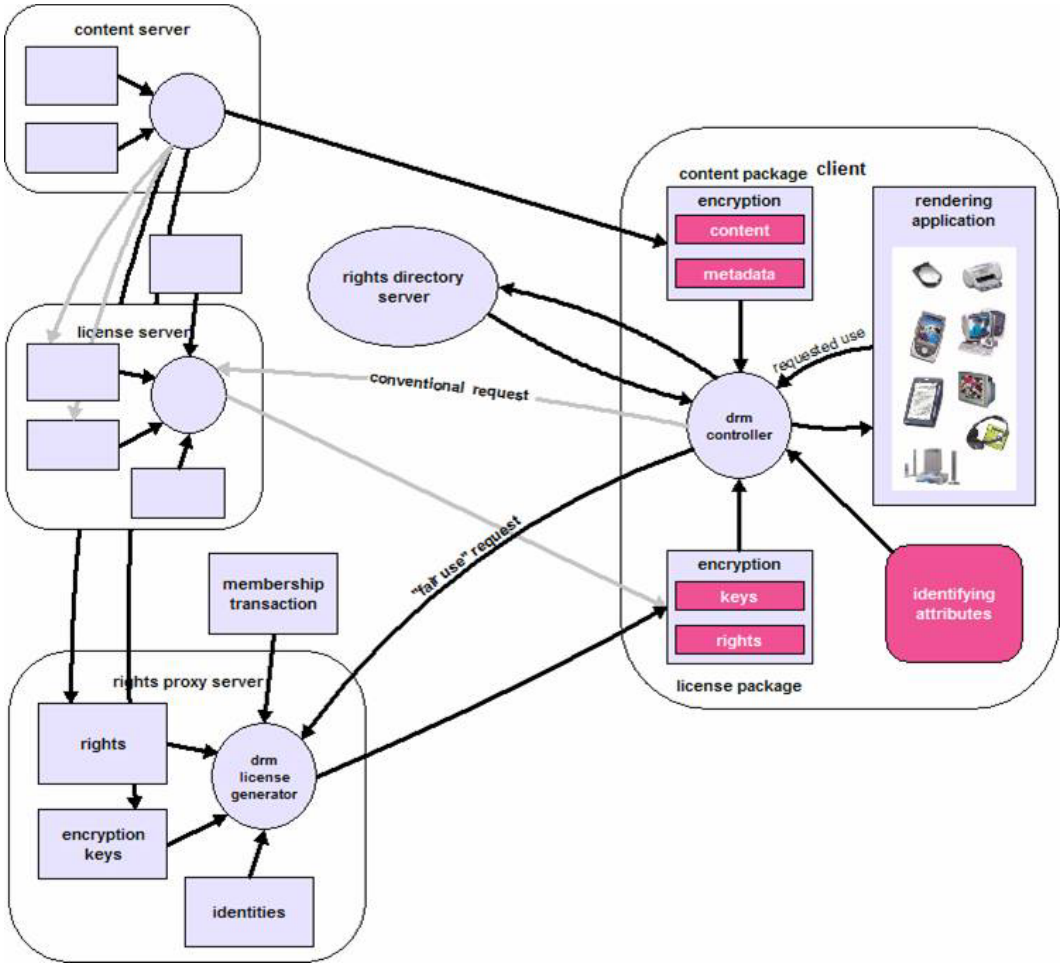
A rights expression language does not have to assume any enforcement. This is the case with the simple REL developed by the proponents of the CreativeCommons [9]. The CreativeCommons web site provides a step-by-step license creation mechanism for writers, musicians and artists who make their works available over the Internet. It is a short statement of ownership and the creator's instructions regarding reuse of all or part of the work. Unlike XrML or ODRL, which are specifically designed to be a machine-readable statement that will be interpreted by programs, the CreativeCommons license primarily sees humans as its audience.



Trusted Systems

A rights expression language just expresses; it has no enforcement ability on its own. To gain that enforcement the REL must be used in the context of a system. Since the rights can include distribution of protected works from one party to another in the form of sales or lending, the system must encompass all parties and all means of interaction. And this system must be secure from end to end. It must be what is called a "trusted system."

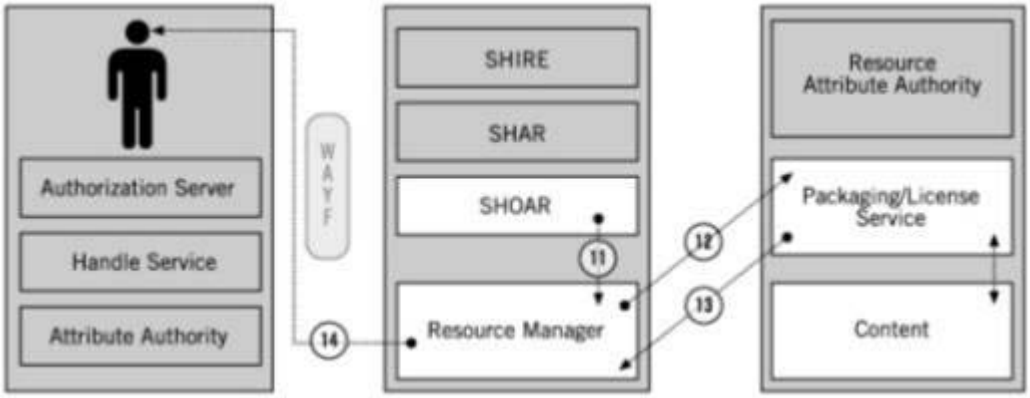
The development of trusted systems is occupying the attention of computer scientists and companies in the computer area. Trusted systems will be necessarily complex. Although there's probably sufficient theory to understand how they should work, creating trusted systems that actually work and are economical is another matter. John Erickson, of HP Laboratories, has spoken frequently of trusted systems and digital rights management, and this diagram is his illustration of the necessary parts and interactions for a trusted system for DRM:



This is a very high level diagram, in that it doesn't give detail, but you can see from it that the interaction for a rights management transaction will require a lot of parts working together. Included in those parts is the computer of the end user, the one on your desk at home, and part of the trusted system is that this computer must be made trustworthy. Trustworthy in this context means that the system must be able to trust that your computer will obey the rules of the rights management software, regardless of what you instruct it to do. As a simple illustration, let's say that you have paid a fee for a book and you can read it for two weeks starting December 1, 2003. Time passes and it's now December 13 and you're not very far along in the book. However, date and time are settings on your computer that you have control over. You can simply reset the clock to December 1 and continue reading. There are a number of ways that this can be "fixed" but the upshot is that timekeeping has to be part of the trusted system and not a function of your individual computer. As a matter of fact, a future operating system for personal computers will have the ability to participate in rights management built in to the operating system itself. This is the system under development at Microsoft, originally called Palladium but now known as the Next-Generation Secure Computing Base. Among other functions, this operating system will not allow you to run non-trusted programs when protected content is present on your computer. This is intended to prevent piracy, because a program that would break the encryption on a protected file simply would not be allowed to run. As one speaker at a DRM conference that I attended

said, trusted computing means that content owners can trust that the computer will obey the DRM instead of you, the computer's owner.

Even without the full scale of trusted computing, systems that will put users together with digital content in a way that respects license agreements will be far from simple. A group of researchers working on a system that will authenticate users for licensed content in a university library setting have come up with this high level diagram [10]. Note that this system does not include any protection of the digital content, only the aspect of getting authenticated users to the digital content:



DRM and the Law

The question that generally arises at this point is: Will I still have fair use rights? The answer is: Yes, copyright law will still recognize your rights to fair use. However, with DRM in place, you may not be able to exercise those rights. DRM is not an implementation of copyright law, it is a system for the protection of digital works. While the debate over fair use and first sale rights for digital materials goes on, most creators of DRM systems are careful to avoid any reference to copyright law in their products. DRM will implement licenses through software controls. The rights or grants in these licenses will not look like the rights we have under copyright law; they will look like the grants that can be expressed in a computer environment. A particular license may allow you to make copies of up to five pages in a book. If you need to copy six pages, and if you feel that the copying of those pages would be allowed under the doctrine of fair use, the software will still only allow you to copy five pages. Where the law allows some flexibility and asks us to make judgments, DRM implementations will be quantitative in nature.

Another important difference between DRM and copyright law, and one that causes me some particular consternation, is this: copyright law sets down a few rules about copying and performances. It gives the exclusive right to make copies to the rights holder. It sets out some exceptions to that right. But in general it doesn't attempt to anticipate every possible use of a copyrighted work. A digital rights management system functions in exactly the opposite way. Where copyright law is an expression of "everything that is not forbidden is permitted," DRM takes the approach of "everything that is not permitted is forbidden." Under a system of DRM, any action you wish to take, such as being able to print from a work, has to be explicitly granted. If there is no stated right to print, then the rights management system will not allow printing. This is seen as a necessary

requirement to create secure software by those developing DRM systems, but it has great implications for future uses of protected works. Imagine that a decade or so from today there is some new feature in our computer systems; let's say that computers no longer have screens but instead project displays onto any surface. If this isn't recognized by the DRM system under rights that it already allows for a particular digital resource, then it will not be possible to view that resource on this future computer. And that means that it may not be possible to view it at all because a change in technology isn't recognized by the particular controls that are protecting the resource. The future interaction of DRM and innovation could be stifling, both of innovation and of access to certain intellectual resources. In addition, as we saw above with the Microsoft Reader and its lack of a print function, the approach that requires all possible activities to be actively permitted is not compatible with the public domain, where nothing is forbidden.

DRM and Libraries

There is no doubt that DRM has the potential to have a tremendous impact on libraries and how they do their work. Exactly what the impact will be is hard to predict today because of this is a technology in the early stages of its potential development. But it is possible to present some general cautions based on current experience with protected works.

The good news is that there is nothing about DRM that would inherently prevent library lending. As a matter of fact, the systems that have been and are being developed for the sale of works can be transformed into systems for lending, since lending is virtually identical to a short-term sales transaction. We already have lending of digital works in systems like netLibrary's [11] for ebooks and in recently developed systems for libraries by FictionWise [12] and OverDrive [13]. More sophisticated DRM systems may allow libraries to provide additional services beyond lending, such as integrating digital library materials into courseware at educational institutions.

But DRM is likely to provide significant challenges as well, especially in these areas:

Local Control

Rights management systems, especially when embedded in trusted computing systems, will be on the cutting edge of computer technology for at least some time. These systems require strong security end-to-end, from the producer of the digital product to the end user. Because of their technical requirements it is unlikely that a fully trusted digital rights management function will be included in library computer systems, at least not in a way that is affordable to most libraries. This means that the content and the control of the content will remain in vendor systems, and libraries will "outsource" access to the digital materials to these vendors. This is not unlike the situation in libraries today in relation to online databases and digital reference materials, but the impact of this model should be expected to increase as the technology grows in complexity and expense. Implications of this model range from the library's right to archive materials to issues of patron privacy.

Contracts and User Support

With hard copy works, there is one set of rights that pertains to all. A digital rights management system with a fully developed rights expression language could provide a different set of rights for each publication, and if not for each

publication than at least for each publisher. At the extreme, libraries could find themselves negotiating for user rights on a title-by-title basis. More realistically, there will be classes of works with different sets of rights, and classes of users who can exercise different rights. Some amount of time will be spent by library staff mediating between the users and the rights packages, especially as users gain experience with the restrictions imposed by DRM. You can imagine a time when a user comes to the reference desk looking for a book on a topic but specifying that it must be one that allows some printing, or that can be rendered in large type on a particular device. The user support overhead for libraries must be calculated into the cost of purchasing and managing these materials.

Archiving and Future Use

There's an interesting contradiction taking place today when it comes to digital materials. Although some titles are available on a term-limited licensing basis, many titles are being offered for sale to libraries. Sale in this case meaning a permanent acquisition. Sale is what makes sense to libraries, who insist on the ability to purchase electronic materials even if they do not physically acquire the digital files. Sale also makes sense to publishers whose entire business model is based on units sold. But we are not even sure how to archive and provide some guarantee of future access to digital files that have no rights management controls applied to them, and the addition of DRM into this future makes matters much worse. If it takes an entire complex system to allow a user to open and read a book, what happens twenty or fifty or a hundred years from now when that system no longer exists? When the default is that usage rights must be positively granted, a loss of that granting system means that no use can take place. DRM in itself does not make digital archiving impossible, but it does compound the problem.

The bottom line is that digital works are in our future, and that digital works need protection because they can be easily copied. This we cannot change. But librarians can have an impact on the development of DRM technologies by participating in the discussions taking place in standards organizations and the research arena. It is our professional duty to take part in the development of technologies that will affect the future of reading and information access.

References

[1] http://www.law.berkeley.edu/journals/btlj/articles/12_1/Stefik/html/text.html
[2] Adobe Acrobat Reader: http://www.adobe.com/products/acrobat/readermain.html
[3] Microsoft Reader: http:// www.microsoft.com/reader/
[4] Renato Ianella, Digital Rights Management (DRM) Architectures, D-Lib Magazine, v. 7, n. 6, June, 2001, http://www.dlib.org/dlib/june01/iannella/06iannella.html
[5] ODRL home page: http://www.odrl.net
[6] XrML home page: http://www.xrml.com
[7] MPEG-21 standards: http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm
[8] PRISM standard home page: http://www.prismstandard.org
[9] CreativeCommons home page: http://www.creativecommons.org
[10] Martin, Mairéad, et al. Federated Digital Rights Management, D_Lib Magazine, v.8, n. 7/8, July, 2002 http://www.dlib.org/dlib/july02/martin/07martin.html
[11] netLibrary home page: http://www.netlibrary.com
[12] LibWise home page: http://www.libwise.com
[13] OverDrive Digital Library Reserve page: http://www.overdrive.com/library/

Further Reading

OCLC's DRM Readings Page:
http://www.oclc.org/community/topics/rights/basics/default.htm
Electronic Frontier Foundation DRM Page:
http://www.eff.org/IP/DRM/
Law and Technology of DRM Conference Page:
http://www.law.berkeley.edu/institutes/bclt/drm/