AN EXTENSIBLE RIGHTS EXPRESSION PROCESSING SYSTEM

COPYRIGHT NOTICE

[0001] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND OF THE INVENTION

Field of the Invention

[0002] The present invention is directed to rights expression processing systems and methods for processing rights expressions. In particular, the present invention is directed to an extensible grammar based rights expression system and method to allow processing of new rights expressions.

Discussion of the Related Art

[0003] Digital distribution of content by content owners over the Internet is a way to reach out to a potentially large market of Internet users. However, such digital distribution comes with a risk of illegal, or otherwise unauthorized, distribution of the content. Rights Management helps reduce this risk thereby enabling content owners to protect and profit from their digital content. A Rights Management System is utilized to specify the usage rights for content, or other things and to enforce the usage rights. The term "content" is used broadly herein and includes digital works such as music, audio files, text files, books, reports, video, multimedia, pictures, executable code, or any combination thereof.

[0004] Various implementations of Rights Management Systems and rights associated with digital content are known as disclosed by U.S. Patents No. 5,629,980, No. 5,634,012, No. 5,638,443, and No. 5,715,403. Hence, the details of Rights Management Systems are not discussed specifically herein. As evident from these references, a Rights Management system can take many forms, and can employ varying levels of complexity depending on the

security required, the nature of the thing being managed, the complexity of associated rights and conditions, volume and other factors.

[0005] Figure 12 illustrates an example Rights Management system 500 and the associated workflow that can be used to distribute digital content. Typically, when a user goes through an activation process, information is exchanged between activation server 502 and client application 506, and is downloaded and installed in client application 506. Client application 506 serves as a tamper resistant security component and contains the set of public and private keys 504 that are issued by activation server 502 as well as other components such as any necessary engine for parsing or rendering protected content 508.

[0006] The Rights Management system 500 also includes a content preparation application 503 that protects clear content 501 through encryption or other protective mechanism to thereby provide protected content 508. The content preparation application 503 also specifies usage rights in a rights label 510 that is associated with protected content 508. The rights label 510 and specifies usage rights that are available to an end-user when corresponding conditions are satisfied. A rights expression language (herein after "REL") such as XrML™ may be used to specify the rights and conditions set forth in the rights label 510. The rights label 510 and the appropriate encryption key that was used to encrypt the clear content 501 is then provided to the license server 512.

[0007] The license server 512 manages the encryption keys and issues license 514 that allows exercise of usage rights. For example, rights label 510 may include usage rights for viewing protected content 508 upon payment of a fee of five dollars, and viewing or printing protected content 508 upon payment of a fee of ten dollars. Client application 506 interprets and enforces the usage rights that have been specified in license 514 to provide clear content 516 that may be used by the end user.

[0008] The components and modules of the Rights Management system 500 can be located in one or more than one device. For example, the activation server 502 and the license server 512 could be the same server or other device, or plural separate devices. The protected content 508 may be any type of content including a document, image, audio file, video file, etc. Further details of Rights Management systems are set forth in further detail in the references noted above, and consequently, are not discussed specifically herein.

[0009] Thus, Rights Management systems not only protect content, but also enable content owners to manage the sale and use of their content by means of licenses. Licenses

include rights expressions to articulate usage rights and to associate usage rights to a content. Licenses may be specified for different stages during the life cycle of digital content. For example, when digital content is released to a distributor, licenses may be specified by content owners to limit distribution of the digital content to a particular region or a period of time, or to restrict how content may be repackaged. Of course, licenses themselves must be protected as well since they are a controlling facet determinative of how content is used. In this regard, licenses are typically digitally signed by the issuers so that their integrity and authenticity may be verified before being interpreted.

[0010] A license typically includes a grant element, a principal element, a right element, a resource element, and optionally, a condition element. In particular, a license contains one or more grant elements which defines the details of the usage rights granted. The one or more grant elements may specify a principal element, a rights element, a resource element and, optionally, a condition element. The principal element identifies a principal (user) or a group of principals who is/are granted the right to access or use the protected resources, while the rights element names a specific right (e.g. play, view, print, copy) to be given to the principal with regards to accessing or using the protected resources. The resource element specifies the protected resources, and the optional condition element specifies any conditions that are imposed on the right to use the protected resource.

[0011] A license is typically embodied as a rights expression. A rights expression is a syntactically and semantically correct language construct, based on a defined grammar, to convey rights information. As noted, an example of a rights expression language is XrML™. It is important to note that the term "rights expression" as used herein is not be limited to licenses in particular, but refers to any expressions that may be used by the Rights Management system to convey information. Thus, the term "rights expression" and derivatives thereof as used herein generally refers to expression of licenses, license components and/or fragments (such as the grant element, principal element, right element, resource element, and/or condition element), as well as any other appropriate expressions. Moreover, rights expressions may be in a variety of different forms ranging from binary encoded sequences that target resource constrained applications, to multi-level REL construct that describes complex rights information for managed distribution of digital resources and rights granting paradigms.

SUMMARY OF THE INVENTION

[0012] In accordance with one aspect of the present invention, an extensible grammar-based rights expression system for processing one or more rights expressions is provided, the system comprising an interpreter adapted to evaluate the rights expressions, the interpreter including one or more plug-in subcomponents configured to allow evaluation of the rights expressions, a validator adapted to validate compliance with conditions set forth in the one or more rights expressions upon authorization by the interpreter, and a framework adapted provide an interface between the interpreter and the validator.

[0013] In accordance with another aspect of the present invention, an extensible grammar-based rights expression system is provided for processing one or more rights expressions, the system comprising a framework having an extensible architecture with one or more extensibility points for adding extensions to the grammar, and an interpreter adapted to evaluate the one or more rights expressions, wherein the extensions define semantics and syntax of new rights expressions to allow processing of the new rights expressions.

[0014] In accordance with still another aspect of the present invention, a method for processing one or more rights expressions is provided. The one or more rights expressions include a request that requests a usage right associated with a resource for a principal, and a grant that issues a usage right associated with a resource to a principal. The method comprises the steps of registering plug-in components with a framework, making a programmatic call to the framework to authorize requests for usage rights set forth in the one or more rights expressions, finding one or more appropriate plug-in components configured to evaluate each of the requests set forth in the one or more rights expressions, invoking the appropriate plug-in components, evaluating the request against the grant, and returning an authorization result that indicates whether the request is authorized by the grant.

[0015] Yet another aspect of the present invention is in providing a method for processing one or more rights expressions comprising the steps of providing an extensible grammar-based rights expression system having an extensible architecture with an interpreter adapted to evaluate the one or more rights expressions, evaluating the request against the grant using the interpreter, and returning an authorization result that indicates whether the request is authorized by the grant. Preferably, the method further includes the step of adding new extensions to the extensible grammar-based rights expression system to allow processing

of new rights expressions, wherein the extensions define semantics and syntax of the new rights expressions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] Figure 1 is a block diagram of an extensible rights expression processing system in accordance with one embodiment of the present invention.

[0017] Figure 2 is a block diagram of the extensible rights expression processing system of Figure 1 in use.

[0018] Figure 3 is a schematic illustration of a rights expression lifecycle in accordance with one embodiment of the present invention.

[0019] Figure 4 is a schematic illustration of the rights expression generation process stage in the rights expression lifecycle of Figure 3.

[0020] Figure 5 is a schematic diagram illustrating the generation of a rights expression using rights expressions as inputs.

[0021] Figure 6 is a schematic illustration of the rights expression modification process stage in the rights expression lifecycle of Figure 3.

[0022] Figure 7 is a schematic diagram illustrating the modification of a rights expression during the rights expression modification process stage.

[0023] Figure 8 is a flow diagram of a rights expression authorization process stage in the rights expression lifecycle of Figure 3.

[0024] Figure 9 is a flow diagram illustrating the grant validation sub-process in the authorization process stage of the rights expression lifecycle of Figure 3.

[0025] Figure 10 is a flow diagram illustrating the grant resolution sub-process in the authorization process stage of the rights expression lifecycle of Figure 3.

[0026] Figure 11 is a schematic diagram illustrating the grant resolution sub-process of Figure 10 using example data.

[0027] Figure 12 is a schematic illustration of a Rights Management system.

[0028] Figure 13 is a schematic illustration of an item.

DETAILED DESCRIPTION OF THE INVENTION

[0029] Rights Management systems can be applied to digital content and other items, such as services, goods, and the like. For example, rights and conditions can be associated with any physical or non-physical thing, object, class, category, service, or other items for which access, distribution, execution, or other use is to be controlled, restricted, recorded, metered, charged, monitored or otherwise managed in some fashion. Thus, a Rights Management system can be used, for example, to specify and enforce usage rights and condition for any item such as content, service, software program, goods etc. In this regard, to extend the concept of rights management to tangible items, an item ticket can be used to associate usage rights with the item. A license or other rights expression is associated with the item ticket 600, illustrated in Figure 13, with a ticket specification that points to or otherwise indicates the item ticket. The item ticket 600 can be protected with some crypto algorithm or other mechanism for preventing processing or rendering of item ticket 600 except in accordance with an associated license. The item ticket 600, with the security mechanism unlocked, can be a human readable or computer readable coupon, a code, a document, or the like. Accordingly, the phrase "item ticket" refers to any tangible or intangible indication of an item. The item ticket specifies one or more items and thus, usage rights and conditions can be associated with any item including, objects, classes, categories, and services, for which use, access, distribution, or execution is to be controlled, restricted, recorded, metered, charged, monitored, or otherwise managed in some fashion as previously noted.

[0030] As illustrated in Figure 13, the item ticket 600 can be prepared by specifying an item reference 602 and any post license conditions 604 that restrict redemption of the item ticket 600. The item ticket 600 can be linked to the item through the item reference 602. The term "linked" as used herein refers to any type of association, such as a description, a pointer, or the like. For example, the item ticket 600 can include a unique code associated with an item through a database record. When the code is presented to a vendor, the database is searched and the corresponding item can be delivered. Item ticket 600 can also include a human readable description of the item and any post license condition 604 not yet satisfied such as certain location or time the item ticket 600 may be redeemed. Access to the item 600 can be controlled using a license in the manner described above with respect to content. Further details of Rights Management systems utilizing item tickets are disclosed in U.S. application Serial No. 10/159,272 entitled METHOD AND APPARATUS FOR

DISTRIBUTING ENFORCEABLE PROPERTY RIGHTS that was filed on June 3, 2002, the disclosure of which is incorporated herein by reference.

[0031] Regardless of the details of the item, and whether the item is a digital content, an object, a class, a category, a service, or other items, the task of writing valid licenses, and designing and implementing rights expression processing systems is complicated and difficult. Various challenges are encountered in supporting the dynamic aspects of rights expressions and application environments. In particular, rights expressions may be static or dynamic. Static rights expressions are statically defined to describe fixed rights information, and therefore, may not be extended. Dynamic rights expressions however, allow new syntaxes to be added without changing the semantics or grammars of the rights expressions. For example, vendors in the wireless content distribution industry may desire to create a new right, such as a "broadcast" right, to grant Rights Management supported distributors the right to "push" rights-managed contents onto subscribers' devices. Constantly changing technology and business paradigms drive the creation of new types of rights. Without an extensible rights expression processing system, different static systems must be designed and implemented to accommodate new rights expression extensions, variations, and derivations thereof.

[0032] Thus, a rights expression processing system should preferably be extensible to allow accommodation of new rights information that is not currently defined at the time the system is designed and implemented. Consequently, it is desirable to provide an extensible rights expression processing system implemented using an extensible architecture so that it can be used to generate, modify, authorize, and validate rights expressions, whether the rights expressions are static or dynamic without the requirement for a redesign of the rights expression or any underlying schema or data dictionary. As described herein, the extensible architecture of the extensible rights expression processing system allows dynamic processing of extended rights expressions and new rights expressions.

[0033] Figure 1 shows a block diagram of a rights expression processing system 10 in accordance with one embodiment of the present invention that may be used to generate, modify, authorize, and/or validate grammar based rights expressions. As evident from the discussion below, the rights expression processing system 10 is extensible to allow new syntaxes to be added without changing semantics or grammars of existing rights expressions to thereby allow accommodation of new rights expressions. It should be initially noted that

whereas the illustrated embodiment is referred to herein as a rights expression processing system, the present invention may be implemented in any system or device having the architecture described herein. In this regard, the rights expression processing system 10 may be implemented with any type of hardware and software, and may be a pre-programmed general purpose computing device. For example, the rights expression processing system 10 may be implemented using a personal computer, a portable computer, a thin client, etc. The rights expression processing system 10 may be a single device at a single location or multiple devices at a single, or multiple, locations that are connected together using any appropriate communication protocols over any communication medium such as electric cable, fiber optic cable, any other cable, or in a wireless manner using radio frequency, infrared, or other technologies.

[0034] It should also be noted that the rights expression processing system 10 in accordance with one embodiment of the present invention is illustrated and discussed herein as having a plurality of components which perform particular functions. It should be understood that these components are merely schematically illustrated based on their function for clarity purposes only, and do not necessarily represent specific hardware or software. These components may be hardware and/or software implemented to substantially perform the particular functions explained. Moreover, two or more of these components may be combined together within the rights expression processing system 10, or divided into more components based on the particular function desired. Thus, the present invention as embodied in Figure 1 should not be construed to limit the rights expression processing system 10 of the present invention.

[0035] In the illustrated embodiment, the rights expression processing system 10 includes a framework 12 that allows interfacing of the other components of the rights expression processing system 10. The framework 12 is the infrastructure component of the rights expression processing system 10 that orchestrates the interactions among all the other components of the system and the rights expression-processing functions performed by the various components. Preferably, the framework 12 of the rights expression processing system 10 is rights expression-agnostic, i.e. the framework 12 is not tied to any specific syntax or grammar of rights expressions. Moreover, the framework 12 has an extendible architecture with extensibility points that allows various other components to be added thereto to allow processing of new rights expressions. Of course, in other embodiments, the

framework need not be provided with an extendible architecture with extensibility points, but instead, may be implemented with fixed components.

[0036] The rights expression processing system 10 of the illustrated embodiment also includes a parser 14 adapted to allow parsing and manipulation of data. In particular, the parser 14 is a component that carries out the underlying manipulation operations such as input, output, syntax validation, and manipulation of the rights expression 24 to be processed by the rights expression processing system 10. Preferably, the parser 14 is generic and implemented as a plug-in component that could be replaced by another parser of compatible capability without affecting the behavior of the rights expression processing system 10.

[0037] Again, the term "rights expression" as used herein is not be limited to licenses in particular, but refers to any expressions that may be used by the rights expression processing system 10 to convey information. Thus, the term "rights expression" and derivatives thereof as used herein generally refers to expressions of licenses, license components and/or fragments (such as the grant element, principal element, right element, resource element, and/or condition element), as well as any other appropriate expressions.

[0038] For example, the rights expression 24 may be a grant that comprises a usage right associated with the digital resources and a principal to whom the usage right is granted. The party to whom the grant has been issued is referred to as the "principal", the object of the rights such as an e-book is referred to as the "digital resource", and the rights to use this object the "usage right". Alternatively, the rights expression 24 may be a request that comprises a usage right associated with digital resources, and a principal requesting use of the digital resources. Moreover, the rights expression 24 may include a grant as well as a request. The rights expression 24 may be expressed in any appropriate rights expression language (herein after "REL") such as XML or XrML™. In the preferred embodiment, the underlying parser 14 hides all rights expression-specific syntaxes and grammars through the use of extensible definitions, for example, the XML or XrML™ schema.

[0039] The rights expression processing system 10 of the present embodiment also includes an interpreter 16 adapted to evaluate and/or authorize rights expressions 24, and to provide an authorization result. In particular, the interpreter 16 evaluates the rights expression 24 based on the grant and/or the request set forth in the rights expression 24. The interpreter then may authorize the rights expression 24 without condition thereby authorizing consumption of the digital resources identified in the rights expression 24. Alternatively, the

interpreter 16 may conditionally authorize the rights expression 24 thereby authorizing consumption of the digital resources identified in the rights expression 24 only upon the satisfaction of one or more conditions that are set forth in the rights expression 24. Moreover, the interpreter 16 may not authorize the rights expression 24 so that consumption of the digital resources identified in the rights expression 24 cannot be consumed.

[0040] In accordance with the illustrated embodiment, the interpreter 16 is implemented as a plug-in component having a plurality of plug-in subcomponents 18 which are configured to allow evaluation and/or authorization of different rights expressions 24. In addition, new plug-in subcomponents may be added to the interpreter 16 to allow evaluation and/or authorization of new rights expressions thereby allowing the interpreter 16 to be extensible. Alternatively or in addition thereto, because the rights expression processing system 10 has an extensible architecture, new interpreters may be added to the rights expression processing system 10 that are configured to evaluate and/or authorize new rights expressions.

[0041] The rights expression processing system 10 further includes one or more validators 20 adapted to validate satisfaction of conditions, if any, that may be set forth in the rights expressions being evaluated. In particular, if the above described interpreter 16 conditionally authorizes the rights expression 24, the validators 20 verify the satisfaction of the conditions prior to allowing the consumption of the digital resources identified in the rights expression 24. In the illustrated embodiment of the rights expression processing system 10, validator A through validator M are provided, each validator being configured to evaluate a specific type of condition, by verifying the value of a state variable, for example. The extensible architecture of the rights expression processing system 10 allows new validators that are configured to evaluate and/or authorize new rights expressions to be added to the rights expression processing system 10. For example, a validator N (not shown) implemented as a plug-in component may be added later to allow validation of a new condition that may be expressed in the rights expression to be processed.

[0042] Furthermore, the rights expression processing system 10 also includes a rights expression enabled component referred to herein as an application 22. The application 22 component represents the "user" of the rights expression processing system 10. The user can be a creator, a distributor, or a consumer of rights expressions. Examples of application 22 include rights authoring applications and systems, rendering applications of digital content such as multimedia, video, picture images, and musical artwork, and/or web-service

execution systems, etc. For example, if the application 22 is a consumer of rights expressions such as a digital content and/or service rendering application, the application 22 can utilize various components of the extensible rights expression processing system 10 to perform various processing operations such as parsing, validating, or evaluating rights expressions.

[0043] Figure 2 is a block diagram of the extensible rights expression processing system 10 of Figure 1 which shows processing of a rights expression. The application 22 receives the rights expression 24 that is associated with the digital resource 26 and includes information regarding the digital resource 26. As previously noted, the rights expression 24 may include a grant that comprises a usage right associated with the digital resource 26 and a principal to whom the usage right is granted. Alternatively, the rights expression 24 may be, or also include, a request that comprises a usage right associated with digital resource 26 and a principal requesting use of the digital resource 26.

[0044] The application 22 passes the rights expression 24 with the information regarding the digital resource 26 to the framework 12. The framework 12 then locates the interpreter 16 that is capable of evaluating and/or authorizing the particular rights expression 24, and passes the rights expression 24 to the interpreter 16 for evaluation and/or authorization. The locating of the interpreter 16 that is capable of evaluating and/or authorizing the particular rights expression 24 may be implemented in various different ways. In one embodiment, when the interpreter 16 is provided to the rights expression processing system 10 via the framework 12, for instance, using the application 22, the interpreter 16 registers with the rights expression processing system 10, all the rights expressions of a specific namespace that may be evaluated and/or authorized by the interpreter 16. If the interpreter 16 is requested to evaluate and/or authorize a rights expression, the rights expression is looked up to determine if the rights expression was registered by the interpreter 16 which indicates that the interpreter 16 is capable of evaluating and/or authorizing the rights expression. If the rights expression has not been registered by the interpreter 16 thereby indicating that the interpreter cannot evaluate and/or authorize the rights expression, the interpreter 16 interacts with the framework 12 to locate a different interpreter that has registered the rights expression with the rights expression processing system 10, and passes the authorization request accordingly so that the rights expression can be evaluated and/or authorized. Of course, in other embodiments, other methods may be used to locate the appropriate interpreter for evaluating

and/or authorizing rights expressions, and the above method is merely provided as one example.

[0045] The interpreter 16 evaluates the rights expression 24 using one or more plug-in subcomponents 18 which are configured to evaluate different types of rights expressions. As previously noted, the interpreter 16 provides an authorization result that may authorize the rights expression 24 without condition thereby indicating that the application 22 has a right to consume the digital resource 26 unencumbered. Alternatively, the interpreter 16 may conditionally authorize the rights expression 24 thereby indicating that the application 22 has a right to consume the digital resource 26 only upon the satisfaction of one or more conditions that are set forth in the rights expression 24. Moreover, the interpreter 16 may not authorize the rights expression 24 thereby indicating that the application 22 does not have the right to consume the digital resource 26. Of course, the result of the evaluation by the interpreter 16 is based on the grant and/or the request set forth in the rights expression 24.

[0046] In the example where the rights expression 24 includes both a grant and a request, the evaluation by the interpreter 16 is preferably attained by comparing the grant against the request. In particular, the usage right, the resource, and the principal set forth in the request are compared with the usage right, the resource, and the principal included in the grant. The interpreter 16 can then provide an authorization result based on this comparison. More specifically, the interpreter 16 authorizes the request if the usage right, the resource, and the principal of the request and the grant match one another. The interpreter 16 alternatively conditionally authorize the request if the usage right, the resource, and the principal of the request and the grant match one another, and the grant further includes one or more conditions. Alternatively, the interpreter 16 does not authorize the request if the usage right, the resource, and the principal of the request and the grant do not match one another.

[0047] If the interpreter 16 conditionally authorizes the rights expression 24, the validators 20 allow verification of the satisfaction of the conditions prior to allowing the consumption of the digital resource 26 identified in the rights expression 24. Each validator (validator A to validator M) is preferably configured to evaluate one or more specific types of conditions. Thus, the framework 12 searches and identifies the appropriate validator configured to verify the satisfaction of the condition set forth in the rights expression 24 prior to allowing the consumption of the digital resource 26. The search and identification of appropriate validators may be attained by a method of registering and looking up conditions

in a manner similar to that of locating interpreters as described previously. Of course, other methods may alternatively be used as well.

[0048] As previously noted, the preferred embodiment of the rights expression processing system 10 has an extensible architecture with extensibility points and is capable of being extended to evaluate any present and future newly defined grammar-based rights expressions by implementing the various components of the rights expression processing system 10 as plug-in components. Thus, the interpreter 16 is implemented as a plug-in component having plug-in subcomponents 18 to allow additional interpreters and/or plug-in subcomponents to be added to allow evaluation of new rights expressions that are not initially supported by the rights expression processing system 10. In addition, the validators 20 are also implemented as plug-in components so that additional validators can be readily added to allow processing of new conditions. Thus, the rights expression processing system 10 is preferably readily expandable to process new rights expressions as they are developed in response to new rights granting paradigms and applications.

[0049] For example, Vendor A may launch a web-service intending to be the universal issuer and interpreter for rights vouchers, for example, licenses. One obvious challenge that Vendor A would be facing is that it is almost impossible to build such a universal system to accommodate all the possible rights granting paradigms at the time of the launch because further developments in technology and commerce will likely result in the need for new rights granting paradigms which will entail use of new rights expressions and/or new conditions. However, because the rights expression processing system 10 in accordance with the present embodiment is extensible in the manner described, Vendor A can build the rights expression processing system 10 using the interpreter 16, and validators 20 that process existing rights expressions. As the need for new rights expression and conditions associated thereto arises, Vendor A can build new components to process these new rights expressions. For example, new plug-in subcomponents for the interpreter 16, new interpreters, and/or new validators may be built and added to the rights expression processing system 10 to process the new rights expressions.

[0050] As previously noted, the interpreter 16 is a plug-in component that is adapted to allow evaluation and authorization of rights expression 24 thereby providing semantic meanings to rights expression 24. As an example, Appendix A shows an example license 50 for the rights expression 24. The license 50 grants a principal or other authorized person

holding a valid key, the unlimited right to view, print, and copy the subject "eBook", for a flat fee of $25.99.

[0051] Referring again to Figures 1 and 2, the interactions between the interpreter 16, the framework 12, and the application 22 of the illustrated embodiment of the rights expression processing system 10 are as follows. Initially, the application 22 registers preferably trusted plug-in components with the framework 12 such as the interpreter 16, validator(s) 20, and any other supporting plug-in components. The application 22 then makes a programmatic call to the framework 12 to authorize the request of the rights expression 24. The framework 12 then searches for, and invokes, appropriate plug-in components that are capable of authorizing the request against the grants in the rights expression 24. In particular, the framework 12 identifies the appropriate interpreter 16 and validator 20 that are configured to evaluate and authorize the rights expression 24, for example, by using registration and look-up method as previously described.

[0052] The identified interpreter 16 performs various operations to evaluate the request against what is stated in the grant contained in the rights expression 24. The right, resource, and principal of the request are matched against the right, resource, and principal of the grant. The step of evaluating the grant may also include decrypting the grant if it is encrypted, verifying digital signature if it is signed, and/or authenticating the issuer of the grant. In addition, the step of evaluating the request may include authenticating that the principal is genuine, and verifying the resource. The interpreter 16 returns the authorization results to the application 22, indicating whether the request is authorized, conditionally authorized, or not authorized by the given rights expression 24.

[0053] If the interpreter conditionally authorizes the rights expression 24, the application 22 can then invoke the appropriate validator 20 through the framework 12 to validate the conditions required. This may be attained, for example, by having the application 22 provide context data required for validating the conditions set forth. Using the context data, the validator 20 validates compliance with the conditions set forth in the rights expression 24. If the appropriate context data is not provided, then the conditions may be deem to be not satisfied. Of course, the above described method is merely one example of how the validator 20 can validate compliance with the conditions of the rights expression 24 and in other embodiments, other methods may be used.

[0054] In the rights expression 24 example set forth in the license 50 of Appendix A, the fee element is a condition associated with all of the rights included in the grant to view, print, and copy the eBook. The fee condition states that the application 22 should only be allowed to exercise the rights if, and only if, a fee of $25.99 has been paid. As shown in the example, the validator 20 accesses a payment record service that provides the required context data to validate that this condition has been satisfied.

[0055] Of course, it should be noted that multiple conditions may be set forth in the rights expression which requires multiple validations. In this regard, plurality of validators 20 can validate these multiple conditions set forth in the rights expressions that represent conditions. The framework 12 manages the validators 20, and invokes them one by one based on, for example, the following validation rules:

```
For every condition on the conditions list
    For every validator on the validators list
        Perform condition validation
        If condition is valid, skip to the next by exiting the inner for-loop
        If condition is invalid, then skip to the next validator
    End-for (validators list)
End-for (conditions list)
If all conditions are valid, exit validation process and return a success status
Else exit validation process and return a failure status.
```

[0056] Of course, the above validation rules are merely one example and other rules may be used instead.

[0057] Figure 3 illustrates a rights expression lifecycle 100 in accordance with one aspect of the present invention. As shown, the rights expression lifecycle 100 includes four basic process stages: generation 110, modification 120, authorization 130, and validation 140. Each of these process stages are performed in this sequence to ensure proper processing of enforceable rights expression 104 in accordance with the illustrated preferred embodiment. Of course, in other embodiments however, the sequence and/or the lifecycle itself may be altered.

[0058] As shown in Figure 3, rights data 102 and/or rights expression RE 104 are taken as inputs in the generation process stage 110 and are processed, the processed rights expression RE being indicated by added prime designations as it is processed. In particular, rights expression RE' 114 is created as an output of the generation process stage 110. Rights

expression RE' 114 is then input into the modification process stage 120 which transforms rights expression RE' 114 into RE'' 124. It should be noted that rights expressions RE, RE', and RE'' need not be different, but in certain cases, may be the same if the rights expression need not be altered during one or more of the process stages.

[0059] In the authorization process stage 130, the rights expression RE'' 124 and any supplementary information are received, and the exercising rights described in the rights expression RE'' 124 is evaluated and authorized, for example, in the manner described above relative to Figures 1 and 2. Following authorization process stage 130 is the validation process stage 140 in which authorized rights expression RE'', or a subset thereof, are validated, for example, in the manner described above relative to Figures 1 and 2 to validate compliance with any conditions set forth therein. Thus, by the end of the various process stages of the rights expression lifecycle 100, the rights expression RE 104 has been transformed in to rights expression RE'' 124 and may be again used with new rights data 102 in step 144.

[0060] It should be evident that each of the above described process stages have their own means for extensibility. In this regard, to facilitate illustration and understanding of each process stage clearly, the above example of an "eBook" is used to clarify how each process stage effect the rights expressions, and allow extensibility without the need to modify the core rights expression processing system. For example, a publisher of the eBook may wish to allow anyone holding a valid key that was issued by someone the publisher trusts, the unlimited right to view, print and copy the contents of the eBook, as long as they pay a flat fee of $25.99. While rights expressions as may be expressed using a variety of RELs, one REL may be XrML™. As noted previously, an enforceable set of rights of a license XrML™ can be found in the rights expression 24 as set forth in the license 50 of Appendix A. However, as previously noted, the term "rights expression" as used herein refers to any rights expression including fragments of licenses that are not enforceable by itself and should not be interpreted to mean enforceable licenses.

[0061] In order to generate the rights expression for the example eBook in the generation process stage 110, various pieces of information should be made available. For instance, the name of the eBook, the key that identifies the principal(s) wishing to use the eBook, and information regarding the payment service that will process the flat fee of $25.99 for use of the eBook should be made available. In Figures 3 and 4, this information is represented as

rights data 102, Figure 4 showing a schematic diagram of the rights expression generation process stage 110 in the rights expression lifecycle 100 of Figure 3. In particular, the rights data 102 is used by the rights expression processing system 10 to generate the rights expression 114. This rights data 102 may convey this information in either a human or machine readable format.

[0062] Note that the rights expression 104 may or may not be enforceable at this stage of the lifecycle 100. In this regard, the rights expression 104 may simply be a fragment of rights that can be merged with additional data in the next stage, i.e. the modification process stage 120, which together form a rights expression having enforceable rights. It is also possible that the rights data 102 is manipulated into rights expressions 104 manually, for instance, by someone coding the rights expression directly. Whether this process takes place in an automated or manual process, the end result is the generation of rights expression RE' 114, which may be represented in any language or syntax.

[0063] The extensibility of the generation process stage 110 allows addition of rights expressions or manipulation of existing rights expressions without the need to change the core rights expression processing system 10 in any way. As previously noted, one example REL which allows this extensibility is the XrML™ rights expression language based on the W3C XML Schema standard. This standard allows the core schema which defines the language, to remain unchanged, while still allowing external schema references to utilize and expand the language.

[0064] Referring again to the example, if the publisher that offered the eBook wanted to charge the same $25.99 for its use, but was required by law to charge the applicable sales tax for its sale, the publisher, or someone on behalf of the publisher, may add a new rights expression named "tax". This new rights expression would be added into their own schema, referencing the XrML™ core schema as a base. This allows the use all the rights expressions of XrML™ schema along with their new rights expression. The ability to allow manipulation of the rights expressions is most clearly illustrated in Figure 5 which shows a schematic diagram 200 in which a new custom rights expression 204 is combined with a core rights expression 202. The new custom rights expression 204 recites <sx:tax>0.0825</sx:tax> setting forth the rate of tax to be applied. The core rights expression 202 sets forth the fee for the eBook. The combining of the core rights expressions 202 and the custom rights expressions 204 results in the generation of the combined rights expression 206 in which the

fee and the tax is set forth. In the described manner, the rights expression processing system 10 of the present invention can be extended to generate this new combined rights expression 206.

[0065] This extensible capability is made possible by the abstraction of the parser 14 and the other components of the rights expression processing system 10. As shown in Figure 1, the parser 14 is separated from other components of the rights expression processing system 10 by the framework 12. Each component thus communicates with the parser 14 through the framework 12 so that, preferably, direct communication between the various components is not allowed. This allows for a standardized interface, allowing additional components such as the plug-in subcomponents 18 and/or the validators 20 to be added to the rights expression processing system 10 as previously described. Each newly added component could perform its intended action as long as it communicates with the framework 12 in performing its actions.

[0066] Referring to the schematic diagram 200 of Figure 5, a new component such as an interpreter 16 with plug-in subcomponents 18 and/or a validator 20 can be added to the rights expression processing system 10 that understands the tax expression provided in the rights expressions 204, and would work with the parser 14 via the framework 12, to evaluate and/or validate the combined rights expression 206 that incorporates the new "tax" element. The parser 14 would simply query the framework 12 if it knew of a component in the rights expression processing system 10 that can evaluate and/or validate the "tax" element of the rights expression 204. The framework 12 would then locate the required component(s) and ask the component(s) to perform some action such as validating the syntax or any associated data that accompanies it. To the extent that components of the rights expression processing system 10 do not have the capability to process the new "tax" element, components configured to process the new "tax" element can be provided to the rights expression processing system 10. With the new components, the rights expression 206 may be properly processed by the rights expression processing system 10. In this manner, the rights expression processing system 10 is truly extensible to handle proper rights expressions.

[0067] Referring again to Figure 3, the modification process stage 120 of the lifecycle 100 entails taking existing rights expressions in some form, and modifying them in some way to produce a new and/or enforceable set of rights expressions. This is a very desirable feature as will be evident from the description of one example application herein. Referring again to

the example of the eBook, the publisher of the eBook will likely prefer to avoid having to generate a new set of rights expressions for each and every customer. This would be not only time consuming, but tedious as well. One possible solution and an area of extensibility within the modification process stage 120, is for the publisher to generate rights expressions during the generation process stage 110, but leave out the important data that distinguishes one set of enforceable rights from another. Such important data may be, for example, the key that identifies the principal(s) authorized to use the eBook. The location in the rights expressions where the principal is designated could contain a placeholder or a "token" that would allow data to be merged with the rights expressions, resulting in enforceable rights expressions.

[0068] In particular, Figure 6 shows a schematic diagram 300 illustration of one example of the rights expression modification process stage 120 in the rights expression lifecycle 100 of Figure 3. In the illustrated example, the merge data 302 may be the key which identifies "Bob" as the principal, i.e. a person authorized to use the eBook. This key is merged with the incomplete rights expression, i.e. a rights template 304, resulting in a complete set of enforceable rights 306 for Bob. For example, using XrML™, the incomplete rights expressions of the rights template 304 may be set forth in the manner below prior to the data merge during the rights expression modification process 102:

```
<keyHolder>
      <dsig:keyValue>
            <dsig:RSAKeyValue>
                  <cgXrML:CGTOKEN TOKENNAME="<tokenName>" />
            </dsig:RSAKeyValue>
      </dsig:keyValue>
</keyHolder>
```

Once the key data has been merged with the rights template 304, the resulting enforceable rights 306 may resemble:

```
<keyHolder>
      <dsig:keyValue>
            <dsig:RSAKeyValue>
                  <dsig:Modulus>
Idvypjad7XoaYhu9tXAYXaENf8li0VvWHBXvs5nGlySw7exuDOv2olqnNapHtz9OviupZRQ/nEa
1i+6TSRuGsw==
                  </dsig:Modulus>
            </dsig:RSAKeyValue>
      </dsig:keyValue>
</keyHolder>
```

[0069] Figure 7 shows a schematic diagram 330 illustrating another example method for modifying a rights expression during the rights expression modification process stage 120 of

the rights expression lifecycle 100. In particular, the schematic diagram 330 shows a tokenized rights expression 332 which contains a token that allows data to be merged into the tokenized rights expression 332. In the present example, the line that appears in the tokenized rights expression 332 that reads "<cgXrML:CGTOKEN TOKENNAME="<tokenName>" />" indicates that entire line is to be replaced with some real data before an enforceable set of rights can be formed. The replacement value for the token in the tokenized rights expression 332 is provided in the replacement value expression 334 which substitutes for the token in the tokenized rights expression 332. This results in the resultant rights expression 336 which is an enforceable set of rights in the present example.

[0070] In other embodiments, another way that rights expressions may be modified during the modification process stage 120 is by applying digital signature requirements. Digital signatures are often used to ensure that the contents have not been tampered with. The signature can be used to not only identify the person who signed it, but also to ensure the integrity of the data within. Digital signatures are often a vital part of any Rights Management system, but are not mandatory and use of digital signatures (e.g. W3C's DSIG standard) depends on the level of security needed for the system application. Of course, a rights expression may be modified to require such digital signatures to ensure that the contents of the rights expressions have not been tampered with.

[0071] The authorization process stage 130 is the most complex process stage in the rights expression lifecycle 100 of Figure 3, and in one embodiment described herein, entails matching a request to exercise certain usage rights with a set of enforceable rights prescribed in given grant or grants. The authorization process stage 130 in accordance with the present embodiment encompasses various sub-processes which may include grant validation, grant resolution, and grant matching.

[0072] Figure 8 shows a schematic flow chart 400 illustrating an example rights expression authorization process stage 130 in the rights expression lifecycle 100 in accordance with one embodiment. As shown, in flow chart 400, enforceable grants 402 and requests 404 are used in the authorization process stage 130. The requests 404 may include information regarding exercising right 405 desired, identification of an exercising principal 406, and a digital resource 407. The authorization process stage 130 of the present embodiment includes grant evaluation 410, grant resolution 420, and grant matching 430 sub-processes, each of which are described in further detail below. The authorization process

stage 130 results in the request 404 being authorized 470, conditionally authorized 480, or not authorized 490. Of course, it should be appreciated that the rights expression authorization process stage 130 and the sub-processes as described herein are merely examples and the processes may be altered or otherwise modified. For example, the sub-processes of the authorization process stage 130 may be altered to have different process sequence and steps than as shown in Figures 8 to 10 described in detail below. Moreover, each of the sub-processes are also examples only and may be altered to have a different process sequence and steps than as shown. Consequently, the present invention should not be construed to be limited to the examples shown.

[0073] The sub-process of grant evaluation 410 performs a series of tasks to verify critical elements of the grant 402 including, but not limited to, digital signatures, issuer principal(s), digital resources, and exercising right. The sub-process of grant evaluation 410 may be performed by a replaceable component so that rights expressions with new syntaxes and/or semantics could be evaluated. Figure 9 is a schematic diagram illustrating the grant evaluation sub-process 410 in accordance with one example embodiment in the authorization process stage 130 of the rights expression lifecycle 100 of Figure 3.

[0074] In the illustrated example embodiment, grants 402 and a request 404 are evaluated in an element matching step 412 in order to match various elements of the request 404 with the grants 402. In other words, in the element matching step 412, matching elements for various elements such as the right, digital resources, digital identities (such as key identifying the principal) set forth in the request 404 are sought for in the grants 402. If the elements do not match, the grants 402 is deemed invalid in step 416. In the above regard, the rights expression processing system 10 of the present invention is preferably adapted to use plug-ins capable of performing a variety of comparisons to achieve extensibility. Since the element matching step 450 and the supporting sub-components that provide this functionality are implemented as plug-ins, new matching functions could be designed and implemented to support matching of new elements.

[0075] If the elements do match in the element matching step 412, the signature evaluation step 414 is conducted in the present example in which the digital signature of the grants 402 are evaluated to verify that the grant's integrity is intact. This signature evaluation step 414 may also be performed by using a plug-in component. Such a plug-in component may be adapted to also deploy pluggable components to perform digital signature evaluation

and verification functions such as crypto and message digest to maximize extensibility support. If the signature cannot be verified, the grant 402 is deemed invalid in step 416, and in turn, the request is not authorized in step 490 of authorization step 103 as shown in Figure 3. If the signature can be verified, then the grant 402 is deemed valid in step 418 in the present example.

[0076] Referring again to Figure 8, once the grants 402 are deemed valid in step 418, the sub-process of grant resolution 420 is executed in the present example to ensure that all possible elements of the request 404 are matched with the elements of the grants 402. The steps involved in the sub-process of grant resolution 420 of an example embodiment are shown in Figures 10 and 11. Referring to Figure 10, the sub-process of grant resolution 420 includes mutating and matching step 422 in which all permutations of the elements are considered. Further details of the mutating and matching step 422 in accordance with one example are shown in Figure 11 in which the elements of the grants 402 and the request 404 are shown. Referring to Figures 3, 10, and 11, the elements of the requests 404 are expanded during the mutating and matching step 422 so that variations and mutations thereof, are provided in the element set 424. If no match is found, no resolution is provided in step 427, and in turn, the request 404 is not authorized in step 490 of authorization process stage 130 of the rights expression lifecycle 100. If possible matches are found, the instance data binding step 426 is carried out as described below.

[0077] In the instance data binding step 426 of the present example, the variations and mutations of the elements of the request 404 set forth as the element set 424 which may match one or more elements in grants 402 are bound based on the instance data and set apart in the matched and bound element set 428. As shown in the example embodiment of Figure 11, the grant 402 may contain elements where the granted principal is listed as including anyone having the letter 'A' in its identity, followed by anyone identified as "Bob," followed by anyone has the letter 'C' in its identity. The request 404 may contain the exercising principal as the list "Alice", followed by "Bob", followed by "Charles." The resultant permutations of the elements in the request 404 that allow possible matching are shown in element set 424. It should be noted that the element set 424 is not the set of all possible permutations of Alice, Bob, and Charles. Rather, the element set 424 contains only the possible matches with Bob being in the second position, since Bob is not a variable element and must be provided at the second position as set forth in the grant 402. In the present

example, during the instance data binding step 426, each possible match as set forth in the element set 424 are evaluated against the rights expressions "all A", "Bob", and "all C" as set forth in the grant 402 to generate the matched and bound set 428 which is indicated as being bound resolutions 429.

[0078] It should be noted that the example sub-process of grant resolution 420 described does not necessarily require that the elements of the grants 402 have variables that need be resolved for authorization. The sub-process of grant resolution 420 may also be applied to grants that do not require variable resolution as well. It should also be noted that in the preferred embodiment, the component used to perform the above described sub-processes of mutating and matching 422, as well as instance data binding 426, are preferably implemented as plug-in components that can be replaced or added to support new rights expression extensions.

[0079] Referring again to the example embodiment of Figure 8, the sub-process of grant matching 430 is then executed which involves matching each element of the matched and bound set 428 against the evaluated, i.e. bound and completely resolved grant(s) that are derived from the grants 402. A grant is said to be bound and completely resolved when all of the variables and expressions have been instantiated and evaluated with data from the grants 402. If there are no matching elements between the grants 402 and the elements of the matched and bound set 428, then the request is not authorized in step 490 of authorization step 103 of the rights expression lifecycle 100 shown in Figure 3. Of course, the results from the grant matching 430 sub-process may include one or more elements that perfectly match the elements of the grant 402. In such an instance, the request is authorized accordingly. For example, if the matching elements of the request 404 and the grants 402 do not provide any conditions, authorization is issued without conditions as shown in step 470. However, if the grants 402 do provide conditions, a conditional authorization is made as shown in step 480.

[0080] Again, it should be kept in mind that the above described processes of Figure 8 and sub-processes of Figures 9 and 10 are merely provided as examples only and may be modified or otherwise altered in other embodiments to have different process sequences and steps than as shown and described. For example, in the authorization process stage 130, the grant evaluation 410 sub-process having the signature evaluation step 414 may be performed anytime during the authorization process stage 130, e.g. before or after grant resolution 420

sub-process and/or grant matching 430 sub-process. Of course, other modifications to the processes and sub-processes may be made as well in other embodiments.

[0081] Referring again to Figure 3, the validation process stage 104 of the rights expression lifecycle 100 requires the validator 20 to verify the satisfaction of the conditions set forth in the rights expression 104 prior to allowing the consumption of the digital resources identified in the rights expression 104 as previously described relative to Figures 1 and 2. Thus, referring to the eBook example of the XrML™ license 50 set forth in Appendix A, the condition rights expressions placed on the use of the eBook are the verification of the user's digital key, and the payment of $25.99. Both of these conditions have to be validated before the associated rights may be granted to the user. The section of the XrML™ license 50 of Appendix A that defines the required payment is as follows:

```
<sx:fee>
  <sx:paymentFlat>
    <sx:rate currency="USD">25.99</sx:rate>
    <sx:paymentRecord>
      <sx:stateReference>
        <uddi>
          <serviceKey>
            <uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</uuid>
          </serviceKey>
        </uddi>
      </sx:stateReference>
    </sx:paymentRecord>
  </sx:paymentFlat>
</sx:fee>
```

[0082] In this example, the validator 20 processes the $25.99 fee during the validation process stage 140 with a preferred payment service, using the identifier "D04951E4-332C-4693-B7DB-D3D1D1C20844" as designated with the serviceKey sub-element. Once the payment has been authorized, the validator 20 returns a positive result thereby indicating compliance with this condition of the grant. Once all of the other conditions have been validated by the validator 20 in a similar manner, the associated rights set forth in the grant are granted to the user. In this example license 50 set forth in Appendix A, the rights are as follows:

```
<!--The right to play (view) is granted-->
  <grant>
    <cx:digitalWork>
      <cx:locator>
        <cx:nonSecureIndirect URI="http://www.contentguard.com/samples/eBook" Type="URL" />
      </cx:locator>
    </cx: digitalWork >
    <cx:play/>
  </grant>

<!--The right to print is granted-->
  <grant>
    <cx: digitalWork >
      <cx:locator>
        <cx:nonSecureIndirect URI="http://www.contentguard.com/samples/eBook" Type="URL" />
```

```
      </cx:locator>
    </cx: digitalWork >
    <cx:print/>
  </grant>

<!--The right to copy is granted-->
  <grant>
    <cx: digitalWork >
      <cx:locator>
        <cx:nonSecureIndirect URI="http://www.contentguard.com/samples/eBook" Type="URL" />
      </cx:locator>
    </cx: digitalWork >
    <cx:copy/>
  </grant>
</grantGroup>
```

[0083] It should now be apparent in view of the discussion above, that the rights expression processing system in accordance with one embodiment of the present invention provides a novel and advantageous system that may be used to generate, modify, authorize, and validate grammar based rights expressions. As explained, the rights expression processing system may be implemented with an extensible architecture having extensibility points to allow new syntaxes to be added without changing semantics or grammars of rights expressions to thereby allow accommodation of new rights expressions. This extensibility is preferably realized by implementing the components of the rights expression processing system such as the interpreter and validator as plug-in components.

[0084] While various embodiments in accordance with the present invention have been shown and described, it is understood that the invention is not limited thereto. The present invention may be changed, modified and further applied by those skilled in the art. Therefore, this invention is not limited to the detail shown and described previously, but also includes all such changes and modifications as defined by the appended claims and legal equivalents.

Appendix A

An example rights expression that grants to a principal or other authorized person the unlimited right to view, print, and copy the subject "eBook," for a flat fee of $25.99:

```
<license>                                                    50
  <grant>                                                    ↙
    <grantGroup>
      <dsig:keyValue>
        <dsig:RSAKeyValue>
          <dsig:Modulus>
Idvypjad7XoaYhu9tXAYXaENf8li0VvWHBXvs5nGlySw7exuDOv2olqnNapHtz9OviupZRQ/nEa1i+6TSRuGsw==
          </dsig:Modulus>
        </dsig:RSAKeyValue>
      </dsig:keyValue>
    </keyHolder>
    <!--The right to play (view) is granted-->
    <grant>
      <cx:digitalWork>
        <cx:locator>
          <cx:nonSecureIndirect URI="http://www.contentguard.com/samples/eBook" Type="URL" />
        </cx:locator>
      </cx: digitalWork >
      <cx:play/>
    </grant>
    <!--The right to print is granted-->
    <grant>
      <cx: digitalWork >
        <cx:locator>
          <cx:nonSecureIndirect URI="http://www.contentguard.com/samples/eBook" Type="URL" />
        </cx:locator>
      </cx: digitalWork >
      <cx:print/>
    </grant>
    <!--The right to copy is granted-->
    <grant>
      <cx: digitalWork >
        <cx:locator>
          <cx:nonSecureIndirect URI="http://www.contentguard.com/samples/eBook" Type="URL" />
        </cx:locator>
      </cx: digitalWork >
      <cx:copy/>
    </grant>
  </grantGroup>

  <sx:fee>
    <sx:paymentFlat>
      <sx:rate currency="USD">25.99</sx:rate>
      <sx:paymentRecord>
        <sx:stateReference>
          <uddi>
            <serviceKey>
              <uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</uuid>
            </serviceKey>
          </uddi>
        </sx:stateReference>
      </sx:paymentRecord>
    </sx:paymentFlat>
  </sx:fee>
</grant>
```

We claim:

1. An extensible grammar-based rights expression system for processing one or more rights expressions, said system comprising:

an interpreter adapted to evaluate said one or more rights expressions, said interpreter including one or more plug-in subcomponents configured to allow evaluation of said one or more rights expressions;

a validator adapted to validate compliance with conditions set forth in said one or more rights expressions upon authorization by said interpreter; and

a framework adapted provide an interface between said interpreter and said validator.

2. The system of claim 1, wherein said one or more rights expressions is a plurality of rights expressions.

3. The system of claim 2, wherein said one or more plug-in subcomponents is a plurality of plug-in subcomponents configured to allow evaluation of said plurality of rights expressions.

4. The system of claim 1, further including a plurality of validators adapted to validate compliance with different conditions of said one or more rights expressions.

5. The system of claim 2, further including a plurality of validators adapted to validate compliance with different conditions of said plurality of rights expressions.

6. The system of claim 1, wherein said one or more rights expressions includes a grant that includes a usage right associated with a resource and a principal to whom said usage right is granted.

7. The system of claim 1, wherein said one or more rights expressions includes a request that includes a usage right associated with a resource and a principal requesting use of said resource.

8. The system of claim 7, wherein one or more of said plural rights expressions includes a grant that includes a usage right associated with a resource and a principal to whom said usage right is granted.

9. The system of claim 8, wherein said interpreter is adapted to evaluate said request based on said grant, and at least one of:

authorize said request if said requested usage right is issued in said grant;

conditionally authorize said request if said requested usage right is issued with one or more condition set forth in said grant; and

not authorize said request if said requested usage right is not issued in said grant.

10. The system of claim 9, wherein said request is conditionally authorized so that said usage right requested in said request is authorized contingent on one or more condition set forth in said grant.

11. The system of claim 10, wherein said validator validates compliance with said one or more condition set forth in said grant.

12. The system of claim 8, wherein said interpreter is adapted to evaluate said one or more rights expressions based on said request and said grant.

13. The system of claim 12, wherein said one or more rights expressions are evaluated by said interpreter by comparing said usage right, said resource, and said principal included in said request with said usage right, said resource, and said principal included in said grant.

14. The system of claim 13, wherein said interpreter is adapted to at least one of:

authorize said request if said usage right, said resource, and said principal of said request and said grant match one another;

conditionally authorize said request if said usage right, said resource, and said principal of said request and said grant match one another, and said grant further includes one or more conditions; and

not authorize said request if said usage right, said resource, and said principal of said request and said grant do not match one another.

15. The system of claim 6, wherein said interpreter is adapted to evaluate said grant by at least one of decrypting said grant, verifying a digital signature, and authenticating issuer of said grant.

16. The system of claim 7, wherein said interpreter is adapted to evaluate said request by at least one of authenticating said principal, and verifying said resource.

17. The system of claim 1, further comprising an application that interfaces with said framework to provide said one or more rights expressions to said interpreter.

18. The system of claim 17, wherein said application is at least one of a creator, a distributor, and a consumer of said one or more rights expressions.

19. The system of claim 17, wherein said application is at least one of a rights authoring system adapted to author digital content, and a rendering system adapted to render digital content.

20. The system of claim 1, further comprising a parser adapted to manipulate said one or more rights expressions.

21. The system of claim 20, wherein said parser is further adapted to facilitate processing of new rights expressions.

22. The system of claim 20, wherein said parser interfaces with said framework to provide said manipulated rights expressions to said interpreter.

23. The system of claim 1, wherein said interpreter is adapted to be expanded by adding additional plug-in subcomponents thereto.

24. The system of claim 23, wherein said additional plug-in subcomponents are configured to allow evaluation of new rights expressions.

25. The system of claim 23, further comprising additional validators adapted to validate compliance with conditions set forth in said new rights expressions.

26. The system of claim 1, further comprising additional validators adapted to validate compliance with conditions set forth in new rights expressions.

27. A method for processing one or more rights expressions including a request that requests a usage right associated with a resource for a principal, and a grant that issues a usage right associated with a resource to a principal, said method comprising the steps of:

registering plug-in components with a framework;

making a programmatic call to said framework to authorize requests for usage rights set forth in said one or more rights expressions;

finding one or more appropriate plug-in components configured to evaluate each of said requests set forth in said one or more rights expressions;

invoking said appropriate plug-in components;

evaluating said request against said grant; and

returning an authorization result that indicates whether said request is authorized by said grant.

28. The method of claim 27, wherein said step of returning said authorization results includes at least one of authorizing said request, conditionally authorizing said request, and not authorizing said request.

29. The method of claim 28, wherein if said step of returning said authorization results includes conditionally authorizing said request in which said requested usage right is authorized with at least one condition set forth in said grant, further including the step of validating satisfaction of conditions set forth in said grant.

30. The method of claim 27, wherein step of evaluating said request against said grant includes at least one of authenticating said principal, and verifying said resource.

31. The method of claim 27, wherein said step of evaluating said request against said grant includes at least one of decrypting said grant, verifying a digital signature, and authenticating issuer of said grant.

32. The method of claim 27, wherein said request includes a usage right associated with a resource and a principal requesting use of said resource, and said grant includes a usage right associated with a resource and a principal to whom said usage right is granted.

33. The method of claim 32, wherein said step of evaluating said request against said grant includes comparing said usage right, said resource, and said principal included in said request with said usage right, said resource, and said principal included in said grant.

34. The method of claim 33, wherein said step of returning said authorization results includes at least one of:

authorizing said request if said usage right, said resource, and said principal of said request and said grant match one another;

conditionally authorizing said request if said usage right, said resource, and said principal of said request and said grant match one another, and said grant further includes one or more conditions; and

not authorizing said request if said usage right, said resource, and said principal of said request and said grant do not match one another.

35. The method of claim 27, further including the step of processing one or more new rights expressions.

36. The method of claim 35, further including the step of adding a new plug-in component configured to allow evaluation of said one or more new rights expressions.

37. The method of claim 36, wherein said one or more new rights expressions include a new request and a new grant, further including the step of using said new plug-in component to evaluate said new request against said new grant.

38. The method of claim 37, further including the step of returning an authorization result that indicates whether said new request is authorized by said new grant.

39. The method of claim 38, wherein if said step of returning said authorization results includes conditionally authorizing said new request in which said requested usage right is authorized with at least one condition set forth in said new grant, further including the step of validating satisfaction of conditions set forth in said new grant.

40. The method of claim 39, further including the step of adding a new plug-in component configured to validate compliance with conditions set forth in said new grant.

41. An extensible grammar-based rights expression system for processing one or more rights expressions, said system comprising:

a framework having an extensible architecture with one or more extensibility points for adding extensions to said grammar; and

an interpreter adapted to evaluate said one or more rights expressions;

wherein said extensions define semantics and syntax of new rights expressions to allow processing of said new rights expressions.

42. The system of claim 41, wherein said interpreter is implemented as a plug-in component.

43. The system of claim 41, wherein said interpreter includes one or more plug-in subcomponents configured to allow evaluation of at least one of said one or more rights expressions.

44. The system of claim 43, wherein said interpreter is adapted to be expanded by adding additional plug-in subcomponents thereto, said additional plug-in subcomponents being configured to allow evaluation of new rights expressions.

45. The system of claim 41, wherein said one or more rights expressions includes a grant that includes a usage right associated with a resource and a principal to whom said usage right is granted, and a request that includes a usage right associated with a resource and a principal requesting use of said resource.

46. The system of claim 45, wherein said interpreter is adapted to evaluate said one or more rights expressions by comparing said usage right, said resource, and said principal included in said request with said usage right, said resource, and said principal included in said grant.

47. The system of claim 46, wherein said interpreter is adapted to at least one of:

authorize said request if said usage right, said resource, and said principal of said request and said grant match one another;

conditionally authorize said request if said usage right, said resource, and said principal of said request and said grant match one another, and said grant further includes one or more conditions; and

not authorize said request if said usage right, said resource, and said principal of said request and said grant do not match one another.

48. The system of claim 41, wherein said one or more rights expressions include conditions set forth therein.

49. The system of claim 48, further comprising one or more validators adapted to validate compliance with one or more conditions set forth in said one or more rights expressions.

50. The system of claim 49, wherein said one or more validators are implemented as plug-in components, and system further comprises additional validators adapted to validate compliance with conditions set forth in new rights expressions..

51. The system of claim 49, wherein said framework is adapted provide an interface between said interpreter and said one or more validators.

52. The system of claim 41, further comprising an application that interfaces with said framework to provide said one or more rights expressions to said interpreter.

53. The system of claim 51, wherein said application is at least one of a creator, a distributor, and a consumer of said one or more rights expressions.

54. The system of claim 51, wherein said application is at least one of a rights authoring system adapted to author digital content, and a rendering system adapted to render digital content.

55. The system of claim 41, further comprising a parser adapted to manipulate and facilitate processing of said one or more rights expressions.

56. The system of claim 55, wherein said parser interfaces with said framework to provide said manipulated rights expressions to said interpreter.

57. A method for processing one or more rights expressions including a request that requests a usage right associated with a resource for a principal, and a grant that issues a usage right associated with a resource to a principal, said method comprising the steps of:

providing an extensible grammar-based rights expression system having an extensible architecture with an interpreter adapted to evaluate said one or more rights expressions;

evaluating said request against said grant using said interpreter; and

returning an authorization result that indicates whether said request is authorized by said grant.

58. The method of claim 57, further including the step of adding new extensions to said extensible grammar-based rights expression system to allow processing of new rights expressions, wherein said extensions define semantics and syntax of said new rights expressions.

59. The method of claim 58, wherein said step of adding new extensions include the step of adding one or more new interpreters to said extensible grammar-based rights expression system, said new interpreters being adapted to evaluate one or more of said new rights expressions.

60. The method of claim 58, wherein said step of adding new extensions include the step of adding one or more subcomponents to said interpreter of said extensible grammar-based rights expression system, said one or more plug-in subcomponents being configured to allow evaluation of one or more of said new rights expressions.

61. The method of claim 58, wherein said step of evaluating said request against said grant includes the step of using interpreters to match said usage right, said resource, and said principal of said request and said grant.

62. The method of claim 61, wherein said step of returning said authorization results includes at least one of:

authorizing said request if said usage right, said resource, and said principal of said request and said grant match one another;

conditionally authorizing said request if said usage right, said resource, and said principal of said request and said grant match one another, and said grant further includes one or more conditions; and

not authorizing said request if said usage right, said resource, and said principal of said request and said grant do not match one another.

63. The method of claim 62, wherein if said step of returning said authorization results includes conditionally authorizing said request in which said requested usage right is authorized with at least one condition set forth in said grant, further including the step of validating satisfaction of conditions set forth in said grant.

64. The method of claim 63, wherein said step of validating satisfaction of conditions set forth in said grant is attained by one or more validators.

65. The method of claim 64, wherein said step of adding new extensions include the step of adding one or more new validators to said extensible grammar-based rights expression system, said new validators being adapted to allow validation of satisfaction of conditions set forth in one or more of said new rights expressions.