Adding Licensing and Access Control to an Internet Accessible Software Release Management Tool

by

Michael W. Hollis

B.S., Missouri Institute of Technology, 1983

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirement for the degree of

Master of Science

Department of Computer Science

1997

This thesis for the Master of Science degree by

Michael W. Hollis

has been approved for the

Department of

Computer Science

by

____

____

Date__________________

Abstract

This paper describes the design and implementation of a Licensing System and an Access Control System for a Software Release Management Tool. The Software Release Management Tool provides a process through which software can be made available to and be obtained by its users. This tool is designed to automate and optimize the retrieval of a software system's components. Because the tool allows users to retrieve software systems over the internet, some regulation is in order. A Licensing System is used to insure each user agrees to certain terms and conditions prior to retrieving a software system. An Access Control System insures that each user has been given permission to retrieve a software system. These systems work together to regulate the software retrieval process.

Contents

Figures

Introduction

The Software Release Management tool (SRM) is an application that is designed to aide software developers and users in the software release process. This tool was developed by van der Hoek, Hall, Heimbigner and Wolf[1]. Using the Software Release Management tool, a software user can retrieve complex software systems via the internet.

Allowing users to retrieve software over the internet, however, does evoke some concern[2]. By introducing an access control layer into the SRM retrieval process, one can limit the possibility of uninvited software retrieval and usage. Access control can be used to authenticate a user and verify authorization to retrieve the requested software package.

A licensing layer provides a means to display a shrink-wrap license agreement prior to software retrieval[3]. The shrink-wrap license agreement is widely relied upon for mass market distribution of software. Shrink-wrap agreements are unsigned license agreements which state the acceptance on the part of the user is indicated by opening the shrink-wrap packaging or, in our case, by clicking on a button labeled “I agree to the terms and conditions”.

The intent of this paper is to walk the reader through the Software Development Process[4] that was used to develop the licensing and access control enhancements. The first step is to establish the requirements that are needed for SRM’s access control and licensing system. After these requirements have been established, it is time to step back and analyze the problem. The results of this analysis is the input needed for the design of the software. This paper follows with requirements, analysis and design sections to document each of these steps. An implementation section is used to highlight the changes to the design that were needed to successfully implement the enhancements.

Requirements

The first step in the analysis process is to try to determine what features the user would want in a licensing and access control mechanism for SRM. The question that comes to mind is: How will the Licensing and Access Control Systems for SRM be used? I constructed four user scenarios:

1. **No Licensing or Access Control system needed** - This is the easiest scenario to implement. The existing SRM system accommodates this scenario with no improvement. It is important to keep this scenario in mind, however, when implementing a licensing and access control mechanism. There must be an easy way for the user to elect not to have any licensing or access control.
2. **Public Domain Software** - Software released to the public domain by definition do not need to be access controlled. SRM should grant software retrieve permission to all users. Each user that retrieves this software, however, may need to comply by the terms of a licensing agreement. It will be convenient for the user if licensing agreements from dependent public domain systems are combined on a single web page. This way a single click on a confirmation button will allow the user to retrieve all of the public domain software in the dependency graph.
3. **Private access to a group of Software Systems** - A group of users has a collection of proprietary software releases they would like to privately access. This scenario allows for the private beta test of a software system. There is no licensing requirement for access. If a user proves valid identification, access is granted.
4. **Public access to a Software Group with a standard License Agreement for each System within the Software Group** - The software developer would like the public to be able to access all released software, provided each individual has agreed to some terms and conditions. These terms and conditions are constant for each software release that resides within the Software Group.

I believe that the above four scenarios represent the main ways in which licensing and access control functionality will be used within SRM. There are other possibilities, however. Perhaps a group of users would like to have default licensing and global access for their software group. Any licensing and access control system must allow for maximum flexibility.

SRM should not bind users to the confines of one of the situations that I have defined. Licensing and access control must be flexible enough to work with any possible user scenario. By examining situations above, I discovered eight main entities or objects:

Software System - A software entity (program or group of programs) that is submitted to and retrieved from SRM.

Software Group - A set of software systems that logically belong together. The systems may be used together, or they may be used by the same group of customers.

License Agreement - The terms that a user must agree to in order to use a particular Software System or Software Group.

License Agreement Page - A World-Wide-Web accessible document that a customer must read prior to obtaining access to a Software System or a Software Group. A License Agreement Page may require the customer to fill in personal information.

Access Control System - A system used to insure that unwanted access to a Software System or a Software Group is not attained.

Customer - An individual who would like to retrieve a Software System.

UserId - A unique identifier for a customer or group of customers.

Password - A unique sequence of alpha-numeric characters that is known only by a single customer, or by a group of customers. Each UserId will have a Password.

Each of the above entities must interact with each other in a specified manner. It is important to carefully define this interaction prior to designing the system. How will an Access Control System affect a Customer? What will the relationship be between a Software Group and a License Agreement?

The list of requirements below completely defines how the SRM Licensing and Access Control System should function:

- A Software Group will consists of a number of Software Systems.
- Each Software System may optionally have a License Agreement.
- Each Software Group may optionally have a License Agreement.
- Two types of License Agreements will be available:
    - The Standard License Agreement
    - A Custom License Agreement
- If a Software Group has a License Agreement, it becomes the default License Agreement for each Software System within the Software Group. This default agreement may be overridden by creating an individual License Agreement for any particular Software System within the Software Group.
- SRM will present each License Agreement on a web page.
- SRM may optionally combine each Standard License Agreement with other Standard License Agreements on a single web page. The person who submits the Software System or who creates the Software Group can decide whether the Standard License Agreement may be combined with:
    - **all** other Software Systems' Standard License Agreements,
    - only Standard License Agreements from software submitted from the **organization** that submitted this Software System or
    - only Standard License Agreements from software submitted by the **individual** who submitted this Software System
    - **no** other Standard License Agreements.
- Each Standard License Agreement may optionally include any of the following blocks:
    - An Agreement Block for the text of the agreement.
    - A Registration Block so that the user can provide personal information.

- The Custom License Agreement is an alternative to the Standard License Agreement. It is defined solely by the user as a stored web page. The user must write the web page in the HyperText Markup Language (HTML)[4].
- A Custom License Agreement is **not** combinable with any other License Agreements.
- Each Software System may optionally have an Access Control System.
- Each Software Group may optionally have an Access Control System.
- If a Software System or a Software Group has an Access Control System, SRM will only allow access to a user that has a registered UserId and corresponding Password.
- Each Software Group and Software System will optionally have a means for a user to register for a UserId and Password.
- If a user requests registration within a Software System or a Software Group that is accepting new registrations, a standard web page will appear informing the user that SRM has received the request. If the request is approved, a web page will inform the user that SRM will email back a user identification name and password. A system administrator will then need to email the new user's UserId and Password.

Analysis

Object Diagrams

With the help of the Grady Booch methodology[5], I drew a set of object relationship diagrams that include each of these entities. Even though I will not be implementing the SRM improvements in an object-oriented programming language, I still find value in using object-oriented analysis techniques. In Figures 1 through 3, each cloud represents an object or entity. A square on some cloud (hereforth called objectA) represents the "has a" relationship. A line is drawn from this square to a circle drawn on some other cloud (hereforth called objectB). This notation indicates that objectA "has a" objectB. Using Figure 1 as an example, one can say that a Software Group "has a" License Agreement. The numbers on the diagram show quantity. In Figure 1, note that one Software Group (designated by the numeral 1 by the square) has zero or one License Agreements (designated by the 0:1 notation by the circle).

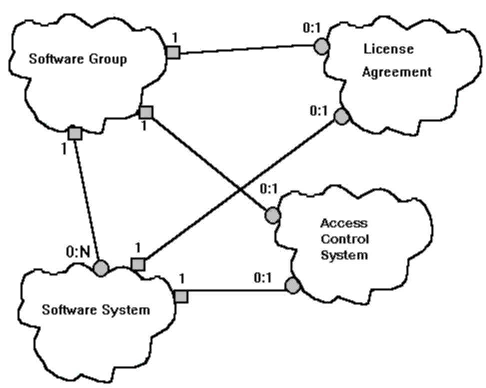


Figure 1: An object diagram of the high level system

Figure 1 shows four of the eight entities discussed in the Requirements section. A Software Group consists of zero to N number of Software Systems. A Software Group has zero or one License Agreements, and has zero or one Access Control System. Likewise, a Software System has zero or one License Agreement, and has zero or one Access Control System.

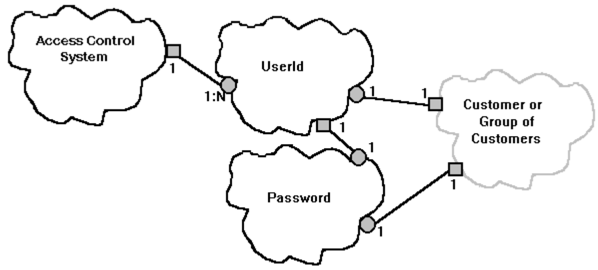


Figure 2: An object diagram of the Access Control System and its children

Figure 2 shows the Access Control System and it's underlying entities. An Access Control System has from 1 to N UserId's. Each UserId has one Password. The diagram shows the Customer or Group of Customers in a lighter shade to indicate that it is an outside influence and is not part of the software. Nevertheless, each Customer or Group of Customers has one UserId and one Password.

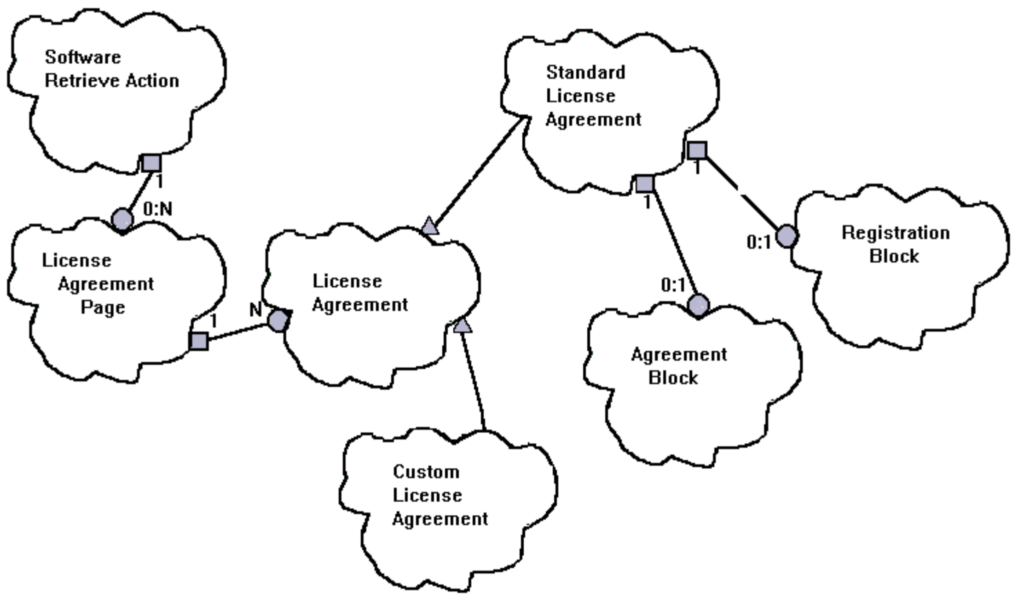


Figure 3: An object diagram of the license page and its children

Figure 3 shows the connection from a Software Retrieve Action to a License Agreement Page to a License Agreement and the entities that each License Agreement has. It also illustrates a new relationship -- the "is a" relationship. In this figure there are two types of License Agreements, a Standard License Agreement and a Custom License Agreement. It is important to note that each

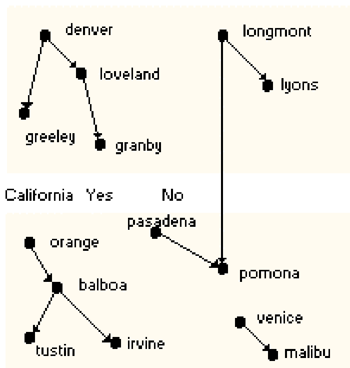
License Agreement "is a" Standard License Agreement or "is a" Custom License Agreement, but not both. A Software Retrieve Action (or a request from a user to retrieve software) will produce 0 to N License Pages. A License Page may have from 1 to N License Agreements. If the License Agreement is a Standard License Agreement it will have zero or one Agreement Blocks and zero or one Registration Blocks. The Custom License Agreement adds more flexibility. If a user does not wish to be confined by using the Agreement Block and Registration Block format, the user may elect to create a Custom License Agreement. This will require writing an HTML coded web page.

Possible Scenarios for Input

This section looks at some of the possible input scenarios. Figure 4 shows two Software Groups, Colorado and California. The left side of the table shows the settings for each Software Group. Note that the "L.A. Combinability" column indicates whether the License Agreement will be combinable with **all** other License Agreements (All), only agreements from the same **organization** (Organization), only agreements from the same **individual** (Individual), or **not** combinable with any other agreement (None). Each Software Group contains a number of Software Systems. The right side of the table shows settings for these systems. The table also shows additional columns to identify who submitted the system, and what organization submitted the system.

Examine the two Software Groups shown in Figure 4. Note that the Colorado group has selected to have a default License Agreement that is combinable with all other License Agreements. Colorado, however, has no Access Control System in place. The California group does have an Access Control System, but elects not to have a default License Agreement.

Group Name	Access Control?	License Agreement?	L. A. Combinability
Colorado	No	Yes	All
California	Yes	No	



System Name	Access Control?	License Agreement?	L.A. Combinability	Who Submitted	What Organization
denver	Yes	No		John	UCB
longmont	No	No	Individual	James	UCB
greeley	No	Yes	Organization	James	UCB
loveland	No	No		Stella	UNC
granby	Yes	Yes	Organization	Simon	UNC
lyons	No	Yes	All	Simon	UNC
irvine	No	Yes	All	Trudy	UC-Irvine
tustin	Yes	No		Trudy	UC-Irvine
orange	No	Yes	Organization	Katie	UC-Irvine
balboa	No	Yes	Individual	Katie	UC-Irvine
pasadena	Yes	Yes	Organization	Ron	UCLA
pomona	No	Yes	None	Tony	UCLA
venice	No	Yes	Organization	Tony	UCLA
malibu	Yes	Yes	All	Tony	UCLA

Figure 4: Possible Input Scenarios

Let us consider five different possibilities of Software System retrieval:

1. **Retrieve "denver" and its dependencies** - in Figure 4 note first that the Colorado Software Group has no Access Control System, so anyone should be able to enter it. The denver system is dependent on three other Software Systems: greeley, loveland and granby. The denver system has an Access Control System in place, and, so does the dependent granby system. Therefore, the user should be expected to enter a valid UserId and Password for both denver and granby's Access Control System prior to successful retrieval. The denver system has no License Agreement, but the Colorado Software Group **has** defined a default License Agreement. SRM must present this default agreement to the user prior to allowing a successful retrieval of the denver system. The greeley and granby Software Systems also have License Agreements that will need to be presented to the user. SRM should not combine the License Agreement for the granby system with denver or greeley's License Agreement. This is because the granby system may only be combined with other systems that were submitted by the University of Northern Colorado (UNC). The greeley and denver Software Systems were submitted by the University of Colorado, Boulder (UCB). SRM **should** be able to combine the greeley and denver License Agreements. This is because the denver License Agreement is combinable with all other License Agreements and the greeley License Agreement is combinable with all other systems' agreements from the University of Colorado, Boulder.

2. Retrieve "longmont" and its dependencies - The longmont system has no License Agreement and no Access Control System. The Colorado Software Group, however, does have a default License Agreement. SRM should present this agreement prior to successful retrieval. The longmont system is dependent on the lyons system from the Colorado group and the pomona system from the California group. Access control is in place for the California group. The user will need to enter a valid UserId and Password for the California group prior to retrieval. The lyons system has defined a License Agreement. SRM should be able to combine this agreement with the default License Agreement needed for the longmont system. The pomona system has also defined a License Agreement. Pomona's agreement, however, is not combinable with any other License Agreements. Therefore, SRM will need to present the License Agreement for the pomona system on a separate web page.

3. Retrieve "orange" and its dependencies - note that the California Software Group has an Access Control System. The user should be required to enter a valid UserId and Password before entering into this group. The California Group does **not** have a default License Agreement. Therefore, each Software System within the California Group is responsible for setting up their own License Agreement. The orange Software System has three dependencies: balboa, tustin and irvine. The tustin Software System is the only one of these with an Access Control System. A user should be expected to correctly enter a UserId and Password for the tustin system prior to successful software retrieval. The orange system has a License Agreement that is combinable only with License Agreements for Software Systems submitted by the University of California, Irvine (UC-Irvine). Since all of the dependent systems have been submitted by UC-Irvine, then SRM should be able to combine all of the License Agreements, correct? No, this is not the case. The License Agreement for the balboa system is restricted. It may only be combined with License Agreements from Software Systems submitted by Katie. While, the orange system was submitted by Katie, the irvine system was not. In this situation, two outcomes are possible. Both of Katie's Software Systems (orange and balboa) could get their License Agreements combined on a single web page. This would leave a separate web page for the irvine License Agreement. Alternatively, the implementation may combine the irvine and orange

License Agreements leaving a separate web page for the balboa License Agreement. Either solution meets the requirements.

4. Retrieve "pasadena" and its dependencies - The pasadena system has an Access Control System in place. A user would therefore need to enter a valid UserId and Password to retrieve pasadena. The pasadena system is dependent on the pomona system. Both the pasadena and the pomona systems have License Agreements. The SRM implementation, however, should not be able to combine these agreements. This is because the pomona system's License Agreement may not be combined with any other agreements.

5. Retrieve "venice" and its dependencies - The venice system does not have an Access Control System in place. It is, however dependent on a malibu system that **has** an Access Control System. Therefore, the user should be expected to enter a valid UserId and Password for malibu prior to successful retrieval. Both venice and malibu have License Agreements. In this case, these agreements should be combinable with each other.

Design

Part One: A Look at the SRM Tool

Figures 5 and 6 show the present and proposed SRM Tool Selection Page and the present SRM Submit Software Page. Figure 5 also shows the additions needed to the SRM Tool Selection Page. Presently, the user creates a Software Group using the separate SRM-ADDGROUP executable. Likewise, the user deletes a Software Group using a separate SRM-REMOVEGROUP executable. These functions should each be integrated in the SRM tool. Integration is necessary because adding a group will no longer be a simple task. The proposed enhancements will require the user to select licensing and access control options when the user creates a new group. Functionality will also need to be added to **modify** the access control and licensing parameters for an existing group. This will be accomplished this via the Modify Group tool.

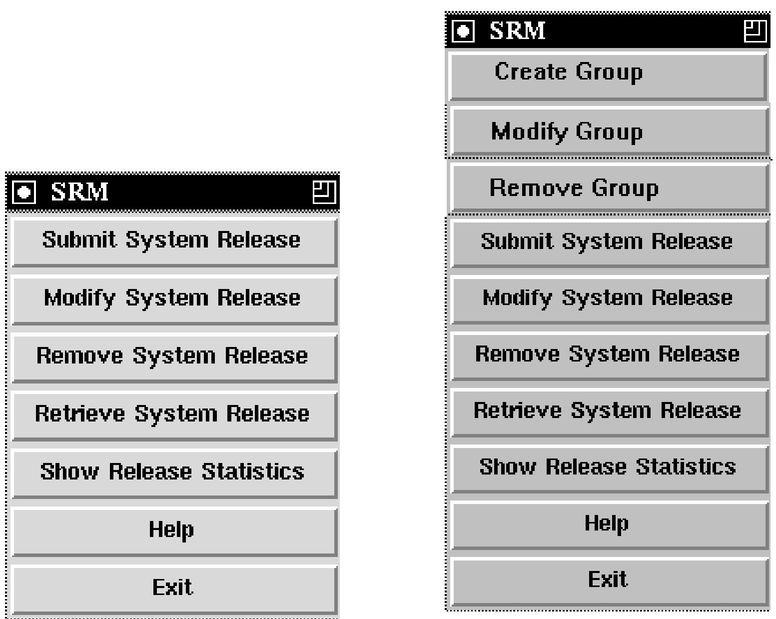


Figure 5: The Present SRM Tool Selection Page and the Proposed SRM Tool Selection Page

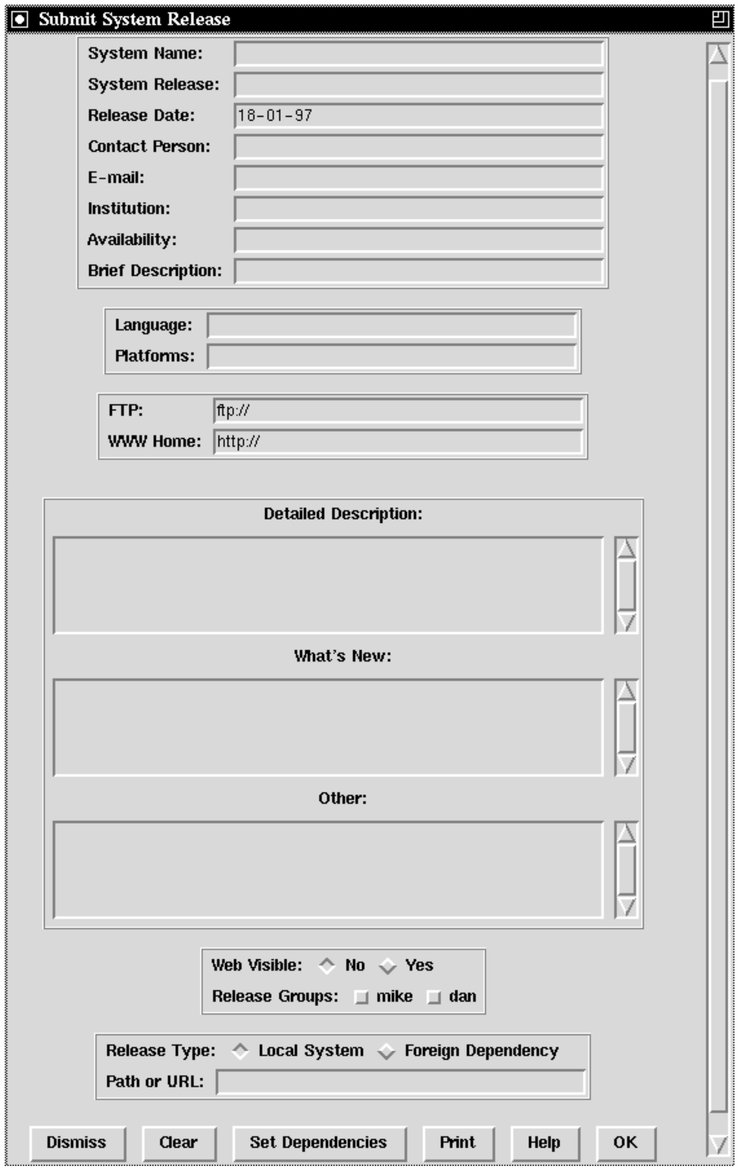


Figure 6: The Present SRM Submit Software Page

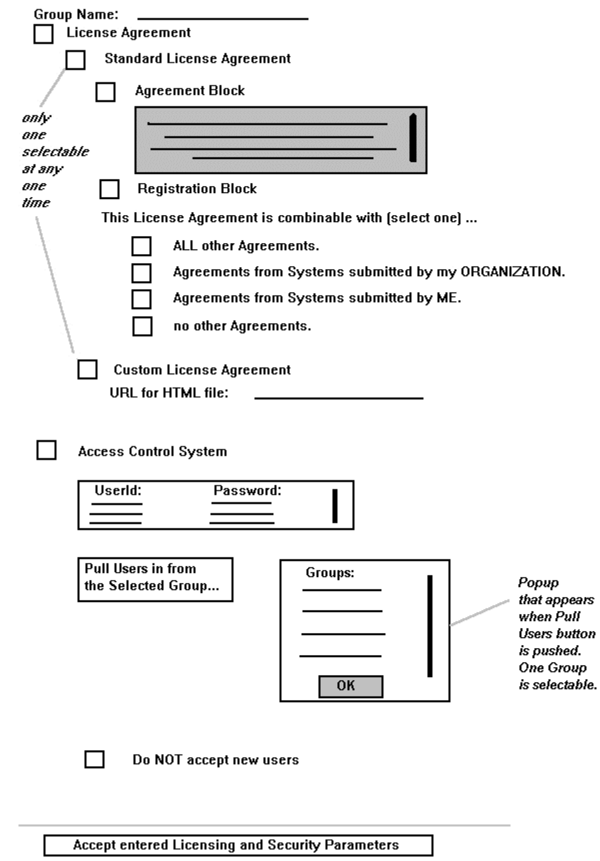


Figure 7: A New Page to Create or Modify a Group

Figure 7 shows the options the user will need to select when the user creates a new group. The implementation will sensitize and desensitize these options according to what the user selects. First, the user will fill in a Group Name. Next, the user decides if a License Agreement is required. If the user clicks on the License Agreement box, the Standard License Agreement will become sensitized and selected. The user can elect to deselect the Standard License Agreement by clicking on the Custom License Agreement box. If the user selects the Standard box, the Agreement Block, Registration Block, and Combinable boxes are highlighted and are selectable. If the user selects the Agreement Block box, the Scrolled Text widget underneath it is highlighted and is editable. The user can select whether a Registration Block will be added to the License Agreement. The user can also select a combine option for the Software Systems' License Agreements. If the user selects the Custom License Agreement, the URL for HTML file field is highlighted and can be edited.

After choosing the License Agreement details, the user can then select if an Access Control System is desired. If the user chooses this box, the scrolled text field with UserId and Password information will appear. The user can now add a list of valid users and their passwords. If the user does not enter any UserId’s or Passwords, an error box will appear informing the user that at least one UserId and Password is needed to implement a valid Access Control System. It is possible to copy a set of UserId's and Passwords from another group by selecting a group from the "Groups:" scrolled list widget and clicking on the "Pull Users in" button. If it is desirable to issue new UserId's and Passwords at a user's request, the user should not select the "Do NOT accept new users" box. If this is not desirable, the user must select the “Do NOT accept” option.

The user will employ this interface to Modify an existing Group. A page will appear asking the user to select which group to modify. After selecting the group, the tool will fill in the information for the selected group into the form shown in Figure 7. Clicking the Accept button will accept the changes made to the licensing and access control policies for the group.

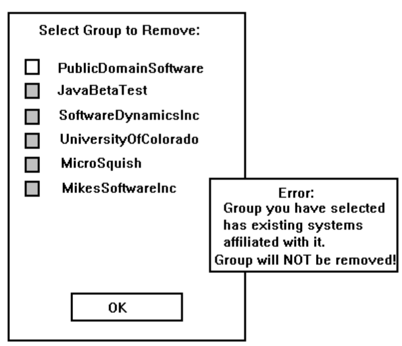


Figure 8: A New Page to Remove a Group

Figure 8 shows the interface needed to remove a group. SRM will not allow a user to remove a group that contains existing System Releases. The existing Modify System Release tool (see Figure 5) will be used to change group affiliations of each release. A user can select the Modify System Release tool to rename the group the Software System is associated with. By renaming each Software System’s group, the user can move all of the Software Systems from one Software Group to another. The user can therefore empty a Software Group prior to removal.

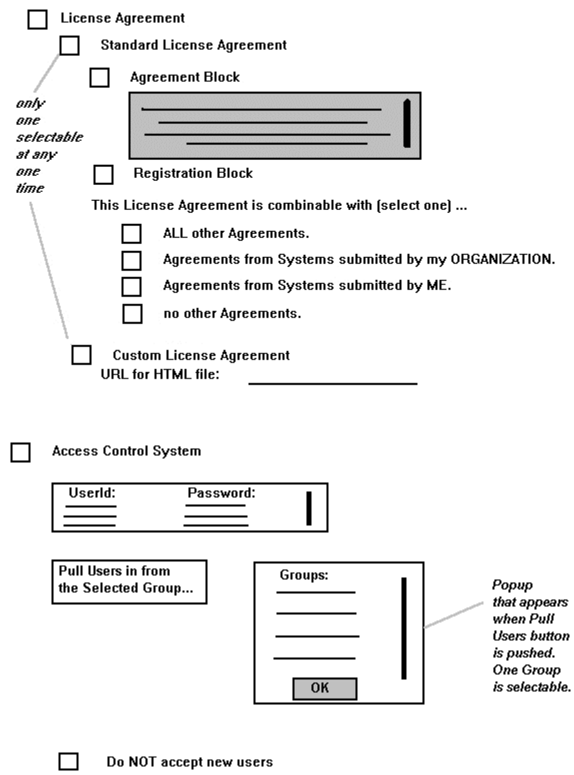


Figure 9: New Additions to the Submit System Release and Modify System Release tools

The interface shown in Figure 9 will be added to the Submit System Release and Modify System Release operations. This interface will have an identical look and operation to the form shown in Figure 7. This form will be inserted after the "FTP: and WWW Home:" fields and prior to the "Detailed Description:" field in the existing Submit System Release interface shown in Figure 6. Likewise, the similar Modify System Release page will have this form added, where the form is filled in with information from the selected release.

Part Two: A Look at the Web Browser Interface

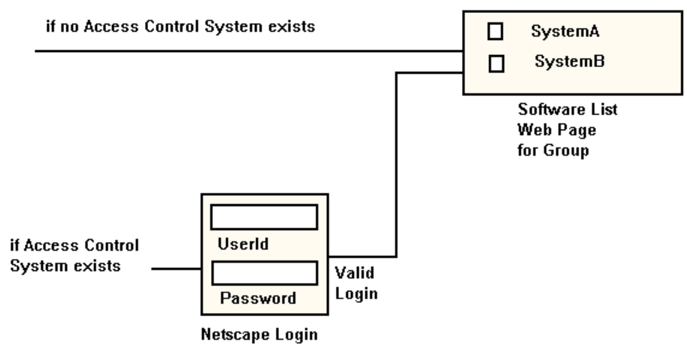


Figure 10: A user-view diagram showing how the system will appear when the User attempts to enter a Software Group

Figure 10 diagrams the interaction as a user attempts to enter a group. The user may directly access the existing Software List Web Page (shown in Figure 18) only if there is no Access Control System for the requested Software Group. If an Access Control System is in place, the Netscape Login Mechanism (shown in Figure 21) will insure proper validation of the UserId and Password. Figure 11 shows additional links from the Software List Web Page.

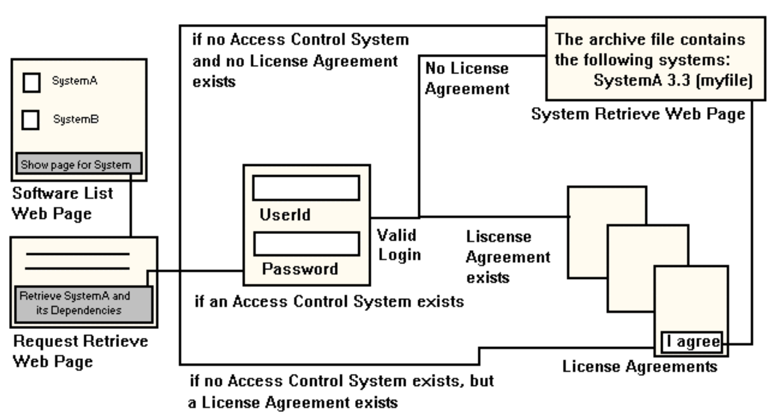


Figure 11: A user-view diagram showing how the system will appear when the User attempts to Retrieve a Software System

Figure 11 diagrams the interaction as a user attempts to retrieve a system and its dependent systems. From the Software List Web Page the user will select a Software System to retrieve. At this point, the existing Request Retrieve Web Page (shown in Figure 19) will appear. Dependencies for the selected Software System may exist. At this point the user may select to retrieve only the initially selected Software System. The user may, however, alternately select to retrieve the initially selected Software System and the set of Software Systems that are dependencies of the initially selected system. From this page, there are four possibilities. In case one, SRM does not need to show any Access Control or License Agreement interfaces for the user to retrieve each of the systems. In this case, the user can directly get to the System Retrieve Web Page. This page confirms that one has retrieved the selected Software Systems. In case two, there are one or more Access Control Systems defined, but there are no License Agreements. Here, the user can retrieve the system after the user has entered a valid UserId and Password for **each** Access Control System via the Netscape Login Mechanism. In case three, there are both Access Control Systems and License Agreements. After entering valid UserId's and Passwords for each Access Control System, the user sees one or more License Agreement Web Pages. In case four, there are no Access Control Systems, but one or more License Agreements exist. In this case, the user proceeds from the Request Retrieve Web Page to the License Agreement Web Page or Pages. After the user has agreed to each License Agreement, the desired Software Systems are retrieved.

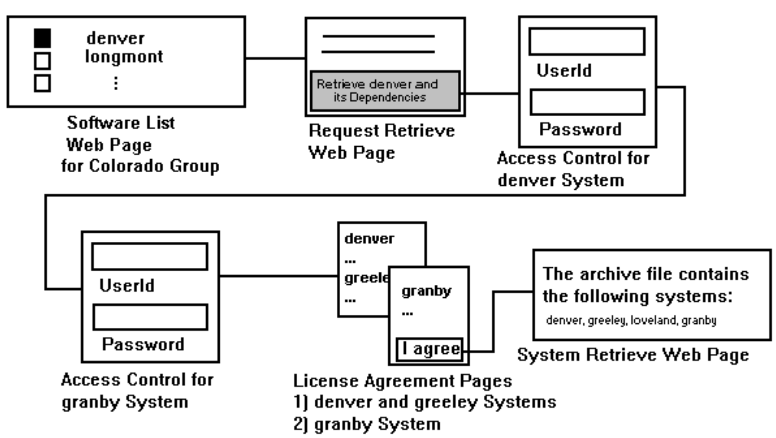


Figure 12: Scenario 1 - Retrieve denver and its dependencies

In the Analysis section of this paper, I defined five possibilities for Software System retrieval. In this section I will examine how each of the five different retrieves work using this design. In the scenario shown in Figure 12, the user attempts to retrieve the denver system and all of its dependent systems. The user selects the denver system from the Software List Web Page. The Request Retrieve Web Page will then appear. In this case, the user selects the "Retrieve denver and its Dependencies" button. Because both the denver and its dependent granby system have Access Control Systems, the user will need to enter UserId's and Passwords for both denver and granby. The License Agreements will appear on two Web pages. The denver and greeley License Agreements are combinable. The granby Agreement will appear on its own page. If the user agrees to the terms of each License Agreement the user proceeds to the System Retrieve Web Page. This page provides confirmation that the user successfully retrieved the desired Software Systems.

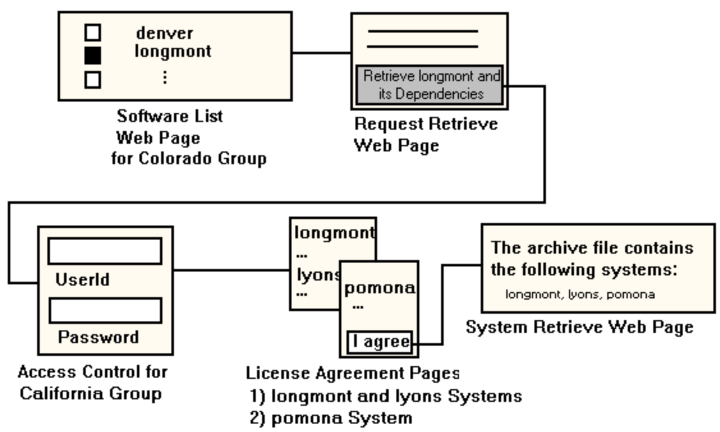


Figure 13: Scenario 2 - Retrieve longmont and its dependencies

In the scenario illustrated in Figure 13, the user would like to retrieve the longmont system, its dependent lyons system, and its dependent pomona system. There is no access control in place for the Colorado group or for any of the Software Systems. There is, however, an Access Control System in place for the California group. The user will need to validly enter a UserId and a Password for the California group prior to retrieving the package. Two License Agreement Pages will appear. The longmont and lyons agreements will be combined. The pomona agreement may not be combined and will therefore have its own page. After the user agrees to the three License Agreements, SRM will retrieve the selected Software Systems.

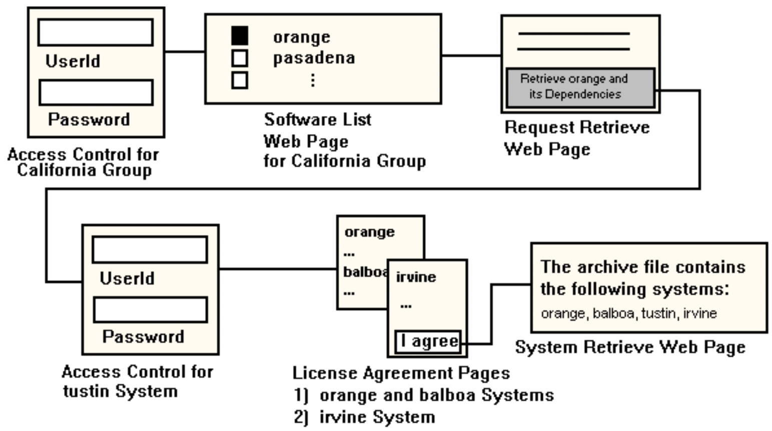


Figure 14: Scenario 3 - Retrieve orange and its dependencies

In Scenario 3, the user is retrieving the orange system and its dependencies. The California Software Group has an Access Control System. Thus, SRM will not allow a user to see the Software List Web Page until the user has entered a valid UserId and Password. The dependent tustin system also has access control defined, so the user must enter a valid UserId and Password for it as well. Two License Agreement Web Pages appear. One combines the orange and balboa systems. A second page appears for the irvine system. If the user agrees to each of the terms in the three License Agreements, the user will then successfully retrieve the orange, balboa, tustin and irvine systems.

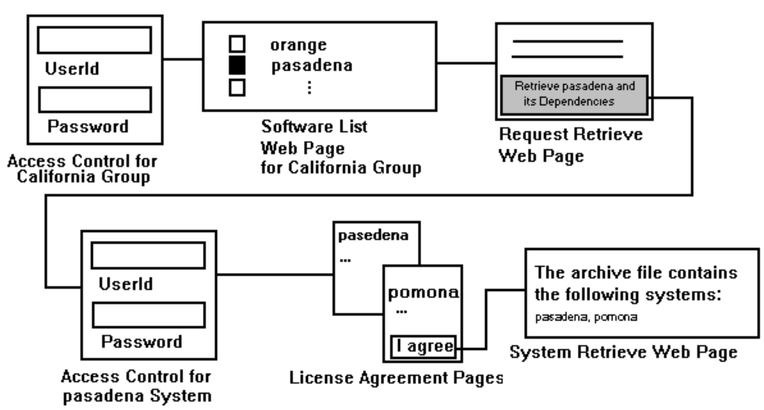


Figure 15: Scenario 4 - Retrieve pasadena and its dependencies

Figure 15 shows the process of retrieving the pasadena system and its dependencies. In this case, the user provides a valid UserId and Password for the California Group Access Control System, and the pasadena Access Control System. The License Agreement for pomona may not be combined with any other agreement. Therefore, the License Agreements for pasadena and its dependent pomona system appear on two separate License Agreement Pages.

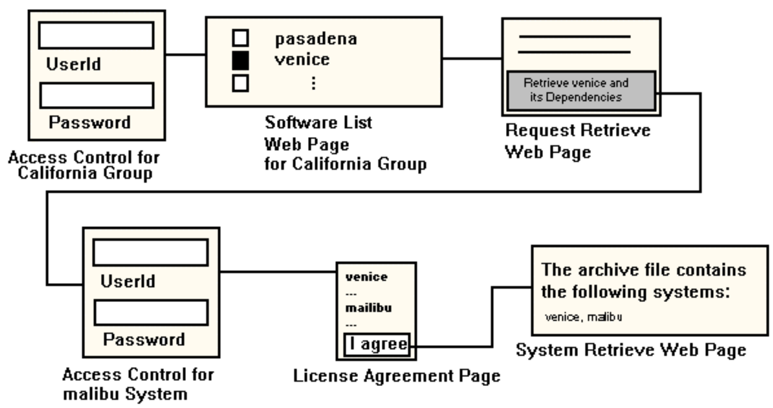


Figure 16: Scenario 5 - Retrieve venice and its dependencies

Figure 16 shows how a user will retrieve the venice system and its dependent malibu system. Access Control Systems for the California Software Group and the malibu Software System are in place. SRM will combine the License Agreements for the venice and malibu Software Systems.

It is important to note that because the proposed system is flexible, a large number of scenarios are possible. By having a flexible system, the implementation can be assured for a maximum number of user situations.

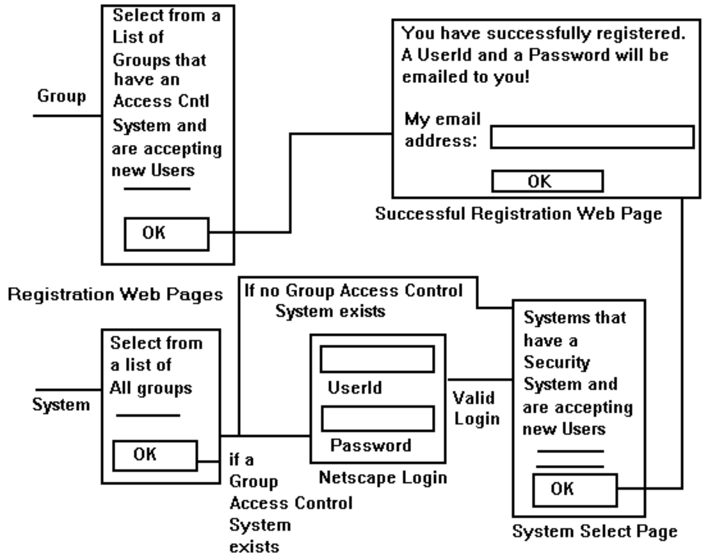


Figure 17: Registering for a Software Group or a Software System

The capability to register a new user can be implemented by creating Software Group and Software System Registration Web Pages. Figure 17 illustrates the links for each of these Web Pages.

If the user is attempting to register for a Software Group, a web page will appear that will allow the user to select a group. SRM must, in this case, limit the group selection. A user would not need to register for a group that does not have an Access Control System. A user would not be able to register for a group that is not accepting new users. If either of these conditions exist for a group, that group will not appear on the Software Group Registration Web Page. Once the user has selected a group and has pressed the "OK" button, the Successful Registration Web Page will appear. This page will prompt the user for an email address. A system administrator will then be responsible for creating a new UserId and Password for the user. The administrator would also need to email the new UserId and Password to the user's entered email address.

If the user is attempting to register for a Software System, the Software System Registration Web Page must allow the user to select from any group. If an Access Control System exists for the group, the Netscape Login will verify the user has a valid UserId and Password prior to advancing to the Software

System Selection Page. The Software System Selection page will only display Systems that have an Access Control System and are accepting new users. Once the user has selected a system and has pressed the "OK" button, the Successful Registration Web Page will appear.

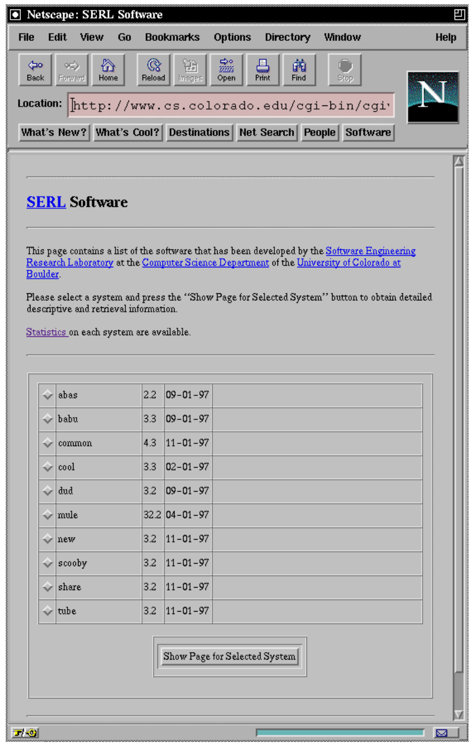


Figure 18: The Existing Software List Web Page

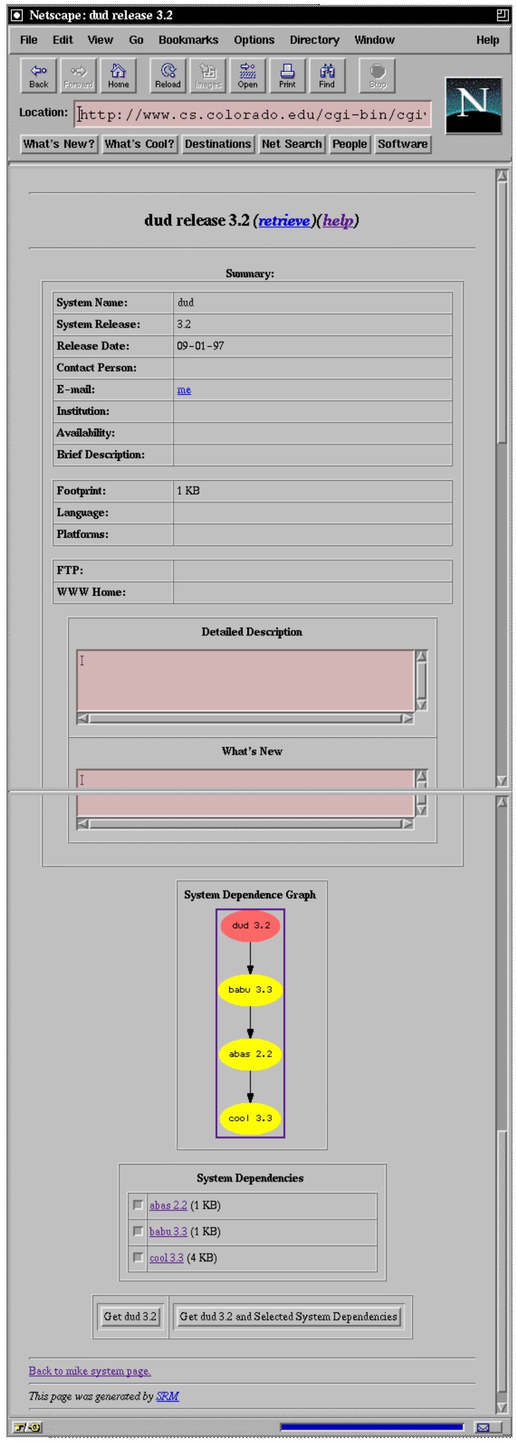


Figure 19: The Existing Software Request Retrieve Page

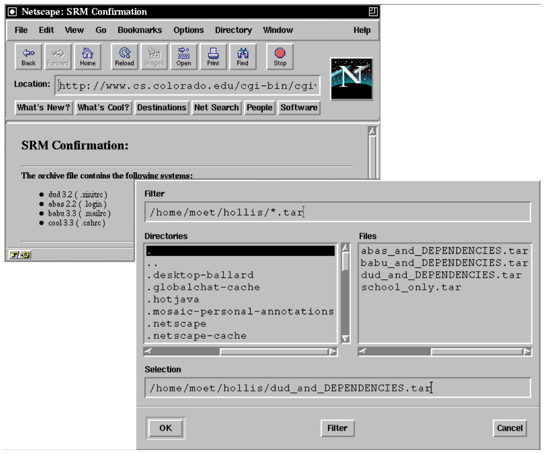


Figure 20: The Existing Retrieve Page

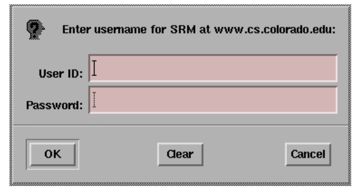


Figure 21: The Netscape Login Mechanism

Figures 18 through 21 are for reference. Each shows more detail of the web pages that appear in the interaction diagrams of Figures 10 and 11.

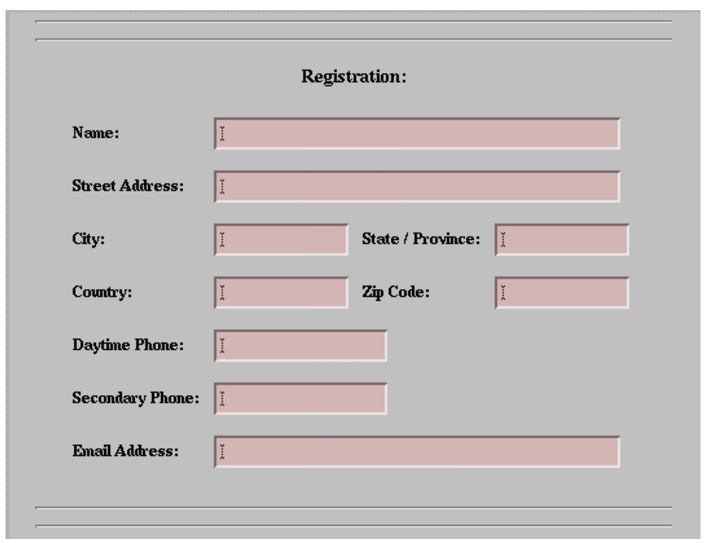


Figure 22: The New Registration Block (may be included in a license page)

Figure 22 shows the Registration Block. This block optionally appears as part of a Standard License Agreement. This block allows a user to enter all of the information needed to register for a Software System.

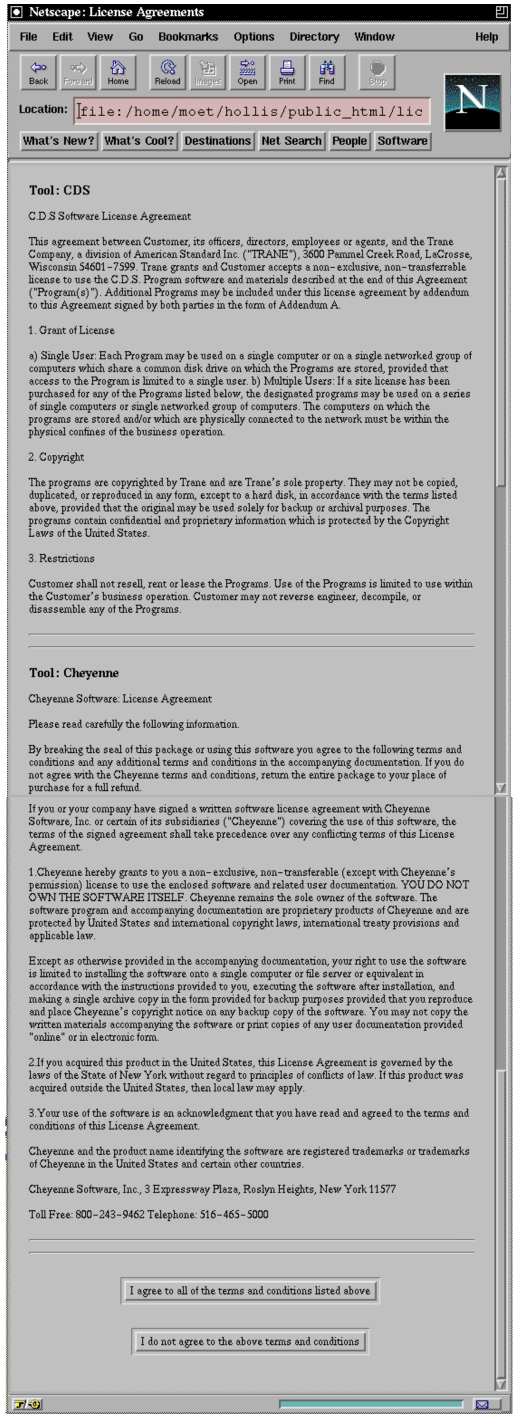


Figure 23: A Sample of a Proposed License Agreement Page with two License Agreements appended -- No Registration Block is used

Figure 23 shows a sample License Agreement Page. Note that two Standard License Agreements have been appended to a single page. By appending Standard License Agreements, the user is given the opportunity to read the agreement for each desired system. Pressing the 'I agree' button confirms that the user agrees to all of the terms in each agreement. Now the user may retrieve several pieces of dependent software with a single click.

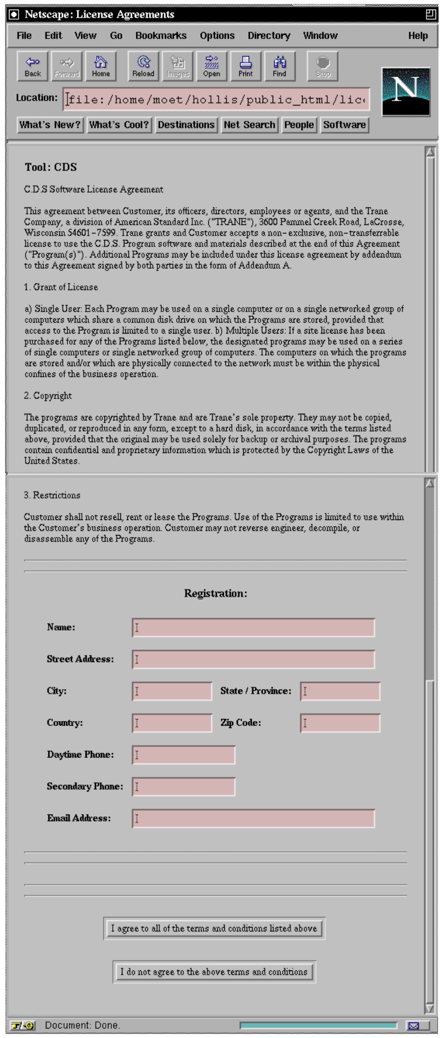


Figure 24: A Sample of the Proposed License Page with Registration Block attached

Figure 24 shows a second sample License Page. In this example, only one Standard License Agreement appears. This agreement contains an Agreement Block and a Registration Block.

Implementation

The implementation of a Licensing and Access Control System for SRM went according to design with a few notable exceptions. Each variation to the design is documented in this section.

Part One: Enhancements to the SRM Tool

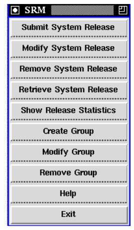


Figure 25: The implemented SRM Tool Selection Page

The new SRM Tool Selection Page looks identical to that planned in Figure 5. The ordering of the buttons was changed. The user will use the System Release tools more often than the Group tools. Therefore, it is more intuitive to have the System Release tools appear above the Group tools.

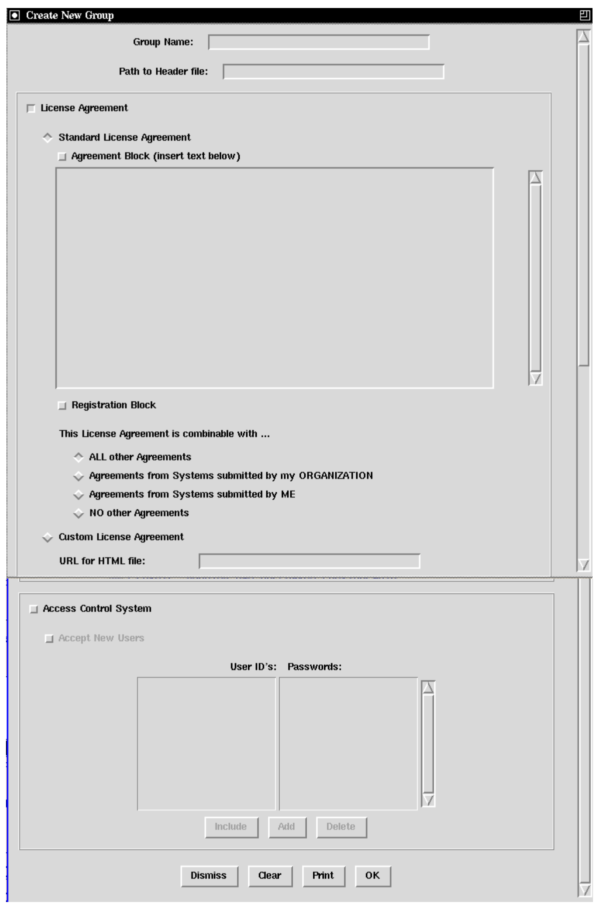


Figure 26: The New SRM Create Group Page

Figure 26 shows the implementation for the SRM Create Group Page. This page is similar to the design shown in Figure 7. Some of the widget highlighting could not be accomplished due to the limitations of the Tcl/Tk programming language[6]. The design also omitted the entry for a Group Header. Input for a header is needed to be consistent with the original SRM Create Group function.

The “Do NOT accept new users” button in Figure 7 was replaced with an Accept New Users button to be more intuitive. The four buttons on the bottom row are consistent with other SRM pages.

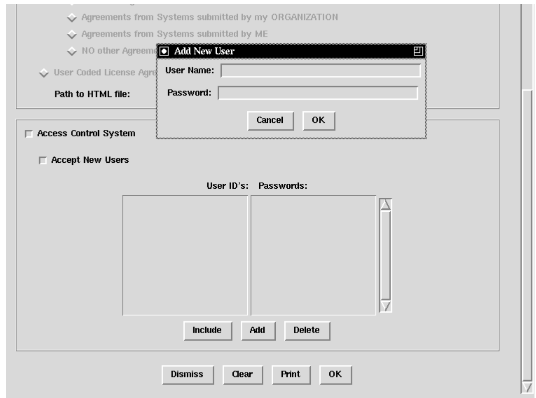


Figure 27: Adding a New User to the Access Control System

Adding, including and deleting UserId’s from the Access Control System is accomplished by using the three button interface shown in Figure 27. The “Add” button will instantiate a popup. The “Add New User” popup will prompt the operator for a UserId and a Password.

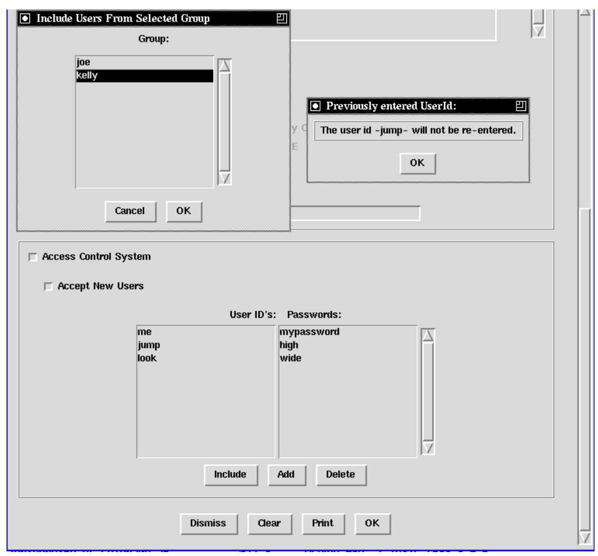


Figure 28: Including Users from an Existing Group

Figure 28 shows the Include operation. Pressing the include button will instantiate a popup to pull in UserId’s from the selected Software Group. Each duplicate UserId will result in a popup instantiation to notify the user that the duplicate entry will not be created.

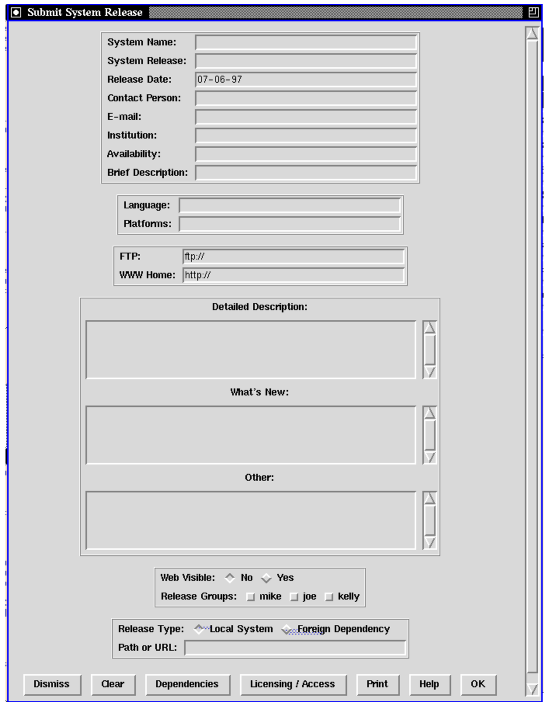


Figure 29: The New SRM Submit Software Page

Figure 29 shows the new SRM Submit Software Page. This page is similar to that shown in Figure 6. The difference is the addition of the "Access / Licensing" button. This button will create a popup which will prompt the user for Licensing and Access Control parameters for the new Software Release. Our original design (shown in Figure 9) indicated that I would insert the Licensing and Access Control parameters into the Software Release page (rather than use a popup). Inserting the Licensing and Access Control selection widgets, however, would have made the SRM Submit Software Page very long. The popup allows the user to more easily ignore Licensing and Access Control if desired. If the Access / Licensing popup is not instantiated, no Access Control System or License Agreement will be installed for the new Software Release.

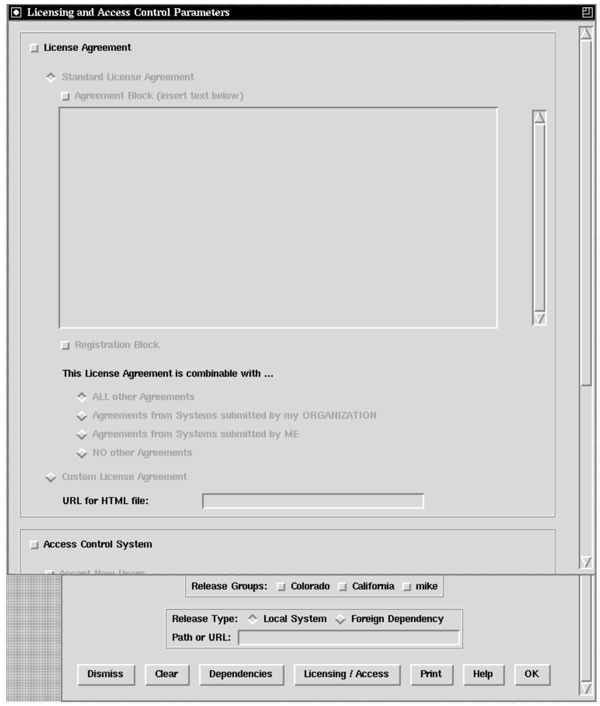


Figure 30: The New Licensing / Access Popup for the SRM Submit Software Page

Figure 30 illustrates the new Licensing / Access popup. The look and feel of this popup is identical to the SRM Create Group Page shown in Figures 26 through 28.

Part Two: Enhancements to the Web Browser Interface

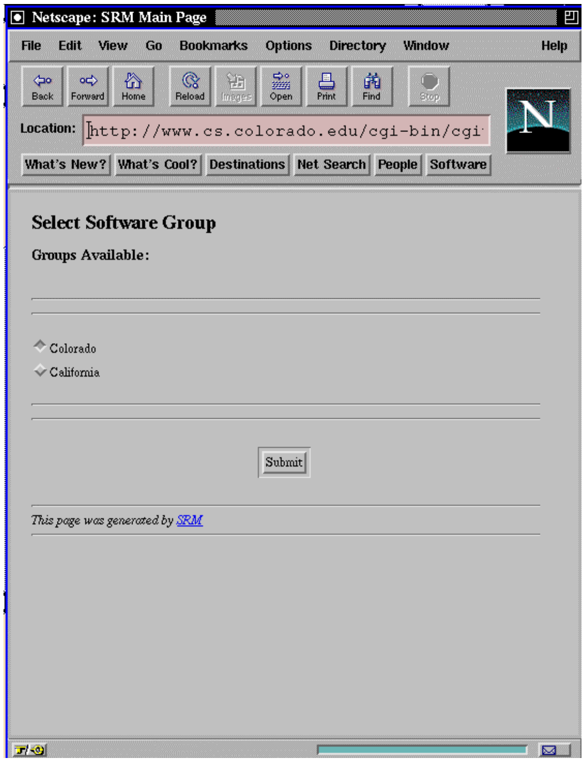


Figure 31: Initial SRM Web Page if no Group Parameter is Entered

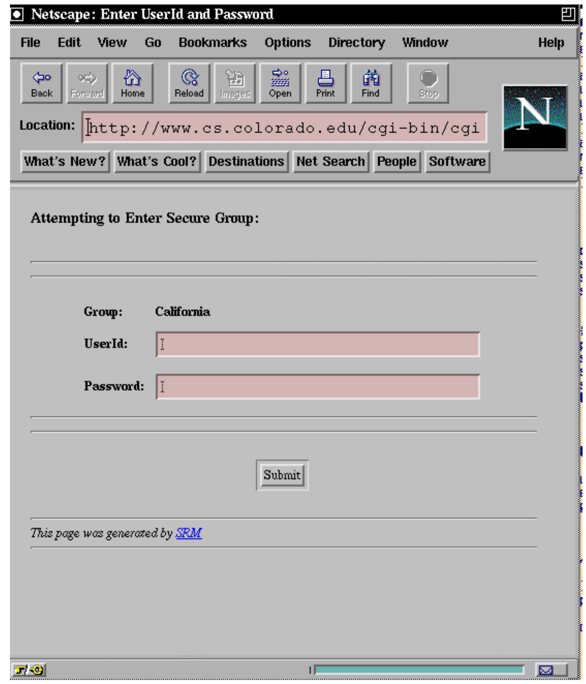


Figure 32: Initial SRM Web Page if a Secure Group Name is Entered

Figures 31 and 32 illustrate the initial SRM Web Page. Implementation of this page depends on the parameters used on the http request. If no GROUP parameter or an incorrect GROUP parameter is entered, the page shown in Figure 31 will appear. This will prompt the user for a correct Software Group name. Pressing the Select button will re-initiate the request with a valid GROUP parameter inserted into the http request.

If the user is attempting to access a secure Software Group and no USERID and PASSWORD parameters are entered, the Web Page shown in Figure 32 appears. This page will prompt the user to enter a valid UserId and Password combination for the requested Software Group. Pressing the Select button will re-initiate the request with the entered USERID and PASSWORD parameters inserted into the http request.

Once valid entry into a Software Group is established, the Software List Web Page shown in Figure 18 will appear. This page allows the user to select the desired Software System for retrieval.

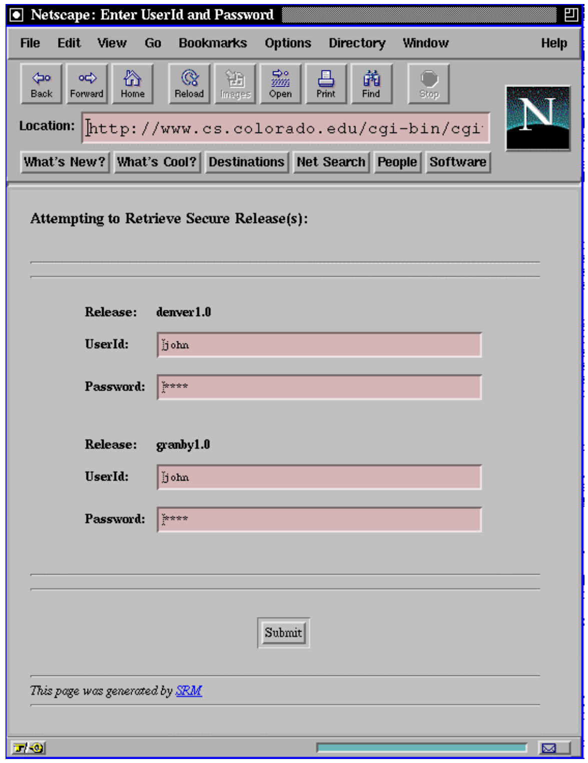


Figure 33: Access Control for Scenario 1 - Retrieve denver and its dependencies

The scenarios for retrieving Software Systems and their dependencies (shown in Figures 12 through 16) were used to test the retrieve implementation. The only change to the design involved the use of the Netscape Login Mechanism. Using the Netscape Login did not allow the flexibility needed for the Software System retrieve operation. There is no way for the Netscape Login Mechanism to examine all of the Access Control systems within a retrieve hierarchy. The alternative was to create the Access Control Web Page shown in Figure 33. Because the Access Control Web Page uses cgi-bin script[7] for its generation, it has the ability to check all of the Access Control systems in advance. This “look ahead” capability is needed when a user is trying to retrieve a hierarchy of Software Systems. Using the “look ahead” feature allows us to pass through each Access Control System prior to retrieving any Software Systems. If a valid UserId and Password for one of the Access Control Systems in the retrieve path can not be entered, the retrieve fails. None of Software Systems in the retrieve path can be obtained.

The Access Control Web Page contains all of the entry needed to retrieve each selected Software System. The Web Page shown in Figure 33 has an area to enter UserId's and Passwords for both the denver and the granby systems. This meets the requirements for scenario 1 shown in Figure 12.

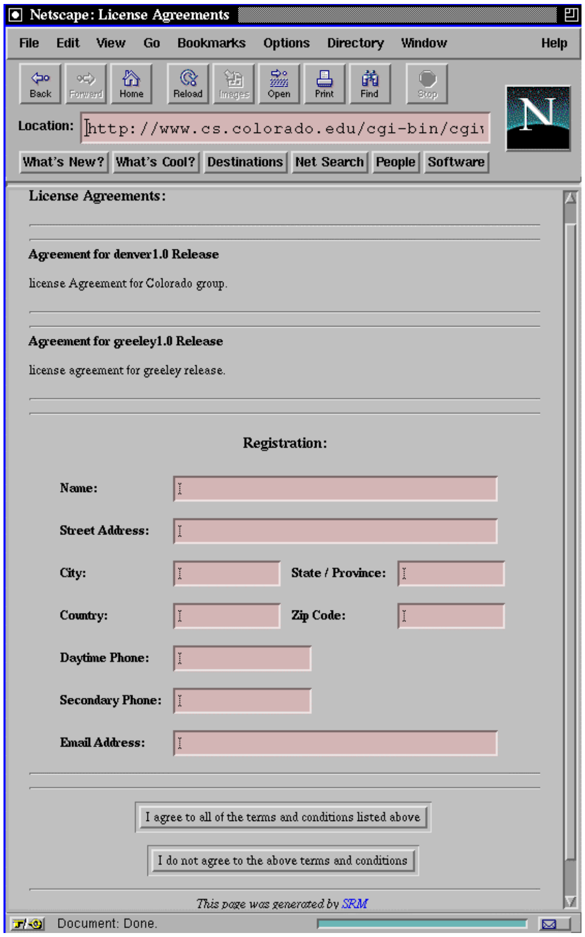


Figure 34: License Agreement Web Page Number One for Scenario 1 - Retrieve denver and its dependencies

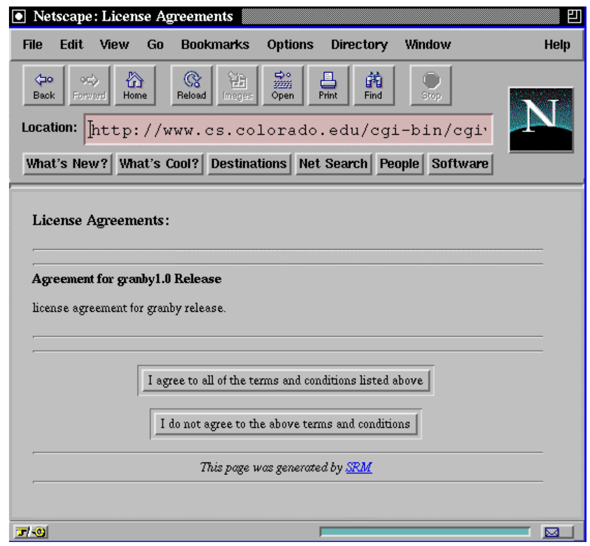


Figure 35: License Agreement Web Page Number Two for Scenario 1 - Retrieve denver and its dependencies

Figures 34 and 35 show the two License Agreement Web Pages for Scenario 1. The separation of Web Pages for each of the scenarios was implemented according to the design. Each scenario was used to verify correct implementation.

Conclusions

The work described in this paper enhances an internet accessible software release management tool. An access control and licensing system was not part the original SRM implementation. Each of these systems, however, is needed for the successful use of SRM over the internet.

Implementing licensing and access control systems to retrieve individual pieces of software is not a complicated problem. Implementing these systems to retrieve a hierarchy of interdependent software system, however, does add some complexity. Integrating these systems into an existing tool also adds a degree of complexity. The use of a clearly defined Software Development Process was of great assistance in breaking down a complicated problem.

The implementation of these enhancements was successful. It is my hope that adding these enhancements will be, in part, responsible for the success of the Software Release Management tool.

Bibliography

[1] Andre van der Hoek, Richard S. Hall, Dennis Heimbigner, and Alexander L. Wolf. Software Release Management. Technical Report CU-CS-806-96, University of Colorado Department of Computer Science, Boulder CO 80309, August 1996.

[2] Dan Farmer, Sun Microsystems, Incorporated. Improving the Security of Your Site by Breaking Into it. Available on the world wide web at *http://www.alw.nih.gov/Security/Docs/admin-guide-to-cracking.101.htm*.

[3] David L. Hayes, Fenwick & West LLP. Shrinkwrap License Agreements - New Light on a Vexing Problem. Available on the world wide web at *http://www.fenwick.com/pub/shrink.html#3*.

[4] Massachusetts Institute of Technology/World Wide Web Consortium. *Hypertext Markup Language - 2.0*, September 1995. Available on the world wide web at *http://www.w3.org/pub/WWW/MarkUp/html-spec/html-spec_toc.html*.

[5] Grady Booch. Object-Oriented Analysis and Design with Applications. Benjamin/Cummings Publishing Company, 1993.

[6] John K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley Publishing Company, 1994.

[7] National Center for Supercomputing Applications. *The CGI Specification*. Available on the world wide web at *http://hoohoo.ncsa.uiuc.edu/cgi/interface.html*.