ICAT 2005 December 5-8, Christchurch, New Zealand

Object and Annotation Authoring in Collaborative Croquet-based Virtual Environments

Julian Lombardi[a], Rieko Kadobayashi[b], Mark P. McCahill[c], Howard Stearns[a], Katsumi Tanaka[b,d], and Alan Kay[e]

[a] Division of Information Technology, The University of Wisconsin-Madison, USA. *{jlombardi,hstearns} @wisc.edu*
[b] National Institute of Information and Communications Technology, Kyoto, Japan. *rieko@nict.go.jp*
[c] Office of Information Technology, University of Minnesota, USA. *mpm@umn.edu*
[d] Department of Social Informatics, Graduate School of Informatics, Kyoto University, Japan. *tanaka@dl.kuis.kyoto-u.ac.jp*
[e] Senior Fellow, HP Labs, Hewlett-Packard Company, Palo Alto, California, USA. *alan.kay@hp.com*

Abstract

To be generally useful for collaborative research and learning, immersive virtual 3D spaces must include intuitive content creation and annotation tools. We describe some initial work to enhance one such environment (Croquet) for use in a broad range of collaborative applications. Annotations in virtual spaces may include features such as comments on objects in the environment, guided tours through the virtual space, and a history mechanism so that the evolution of objects or portions of the space can be replayed and examined. In general terms, an annotation is simply a relationship between one object and another and we examine several methods of displaying these relationships. To extend annotations across communities, the system architecture supports naming and packaging object and meta-information and integration of content into a shared digital repository. By distributing to users the power to create, edit, store, and retrieve objects and annotations, we promote development and re-use of meaningful content. Such systems can have great utility for the development of virtual environments for learning and research.

Key words: Virtual Reality, Croquet, 3D user interface

1. Introduction

The emergence of web-based blogs and wikis highlight the importance of empowering distributed information system users with the capabilities of authoring, editing, and annotating content. What are the most important features of wikis and blogs and how should we apply these to authoring in a 3D collaborative environment?

A blog (or weblog) is a website under the control of one or more users authoring primarily text-based entries displayed in reverse chronological order [1]. Blogs can also be used to store and prepare yet-to-be published entries, so these systems incorporate basic content management and creation tools. Blog popularity has increased as simple and free web browser based publishing tools have become available. Readers of blog entries are generally able to add comments on a blog entry, and in this way a rudimentary community dialog is enabled by way of simple user-written text annotations.

A wiki enables web-delivered documents to be collaboratively created using a web browser with a simple markup language [2]. Wikis are distinguished from blogs by the ease with which pages can be created and updated by multiple users. Many wikis are open to modification without review by either unregistered or registered users, and maintain a history of revisions. The revision history allows users to inspect the evolution of a wiki page and roll back the page to a previous state. For both blogs and wikis, the ability to create, modify, annotate, and store content are important, but the fundamental value is in acting as scaffolding for the development of a collaborative community dialog over time. User-contributed annotation/comments and mechanisms for inspecting the evolution of the dialog over time are capabilities that we seek to carry forward into 3D virtual environments. The need for these mechanisms is clear when considering (for example) creation of virtual museum spaces [3]. Because not all users will simultaneously visit a space, blog/wiki-like annotation functions can augment the virtual experience by allowing for asynchronous user interactions and tours.

As 3D graphics capable mainstream personal computing devices have become affordable, use of virtual reality environments for education and entertainment is attractive [4]. Virtual reality environments provide a (usually immersive) 3D graphics environment [5]. To move beyond simple eye-candy 3D environments, it is necessary to explore how distributed 3D environments can be made available to users as a tool for object creation, editing, storage, and retrieval in support of wide-scale collaborative communication. By implementing blog and wiki functionalities in deeply collaborative and media-rich virtual reality environments it will be possible to empower users in immersive virtual environments with authorship and collaboration capabilities they have come to expect from web-based systems.

A key motivation for this research is the recognition that the wide scale deployment of meaningful 3D environments remains expensive and thus out of reach of most educators. Typical 3D entertainment applications (video games, etc.) require considerable capital expenditures for the development of engaging content. This problem can be avoided if we empower users with intuitive tools to modify open source virtual reality spaces such as those provided by Croquet. By developing content creation tools integrated with a mechanism to persist and share content, the costs of content creation are distributed over a broad social network and broader use of this technology becomes cost-effective.

2. Overall design and features of Croquet

Croquet is a combination of open-source software and peer-to-peer network architecture providing an infrastructure for synchronous real-time problem solving within shared simulations [6]. Croquet supports the collaborative creation, viewing, interaction with and sharing of remote objects via a peer-to-peer network of diverse user hardware (Macintosh, Linux, Windows).

As an ad-hoc multi-user network, Croquet is similar to the web in that users have the ability to create and modify a "home world" and create links to any other such world. Users or groups with appropriate sharing privileges, can visit and work inside other Croquet spaces on the Internet. Croquet's connections between worlds via spatial portals are an analog of web page hyperlinks. Important differences between Croquet and the web are that Croquet is a fully dynamic environment in which everything is a collaborative object, and the Croquet code is modifiable while the system is running. Croquet provides a dynamic, late bound objects environment written in Squeak [7]. By running on the Squeak virtual machine, Croquet objects run bit-identical across hardware platforms. This is a key to avoiding hardware and native operating system incompatibilities.

Croquet provides collaborative capabilities extending to the level of executable code objects. In this way, every visualization and simulation within Croquet is a collaborative - and made from fully modifiable objects. Croquet incorporates replication of computation (both objects and messages/activity). This defines a software development and delivery platform with enormous possibilities as an operating system for both local and global informational resources. By integrating remote console application (such as VNC) into Croquet, the system can incorporate non-collaborative standalone applications including web browsers and other legacy applications into the collaborative space. With these capabilities and the popularity of blogs and wikis for user-created content and comment, the need for a way to add notes and annotations to the virtual space is clear. Before we explore this issue, let us examine Croquet's persistence and digital repository architecture (Figure 1).

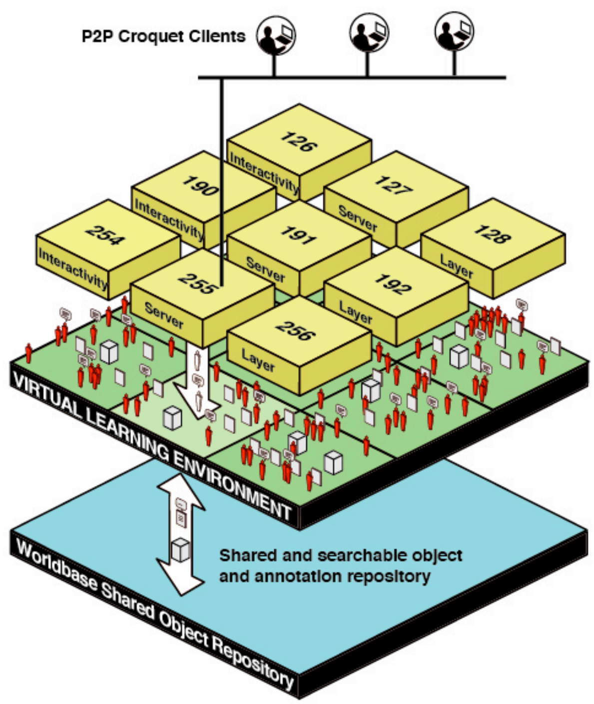

Figure 1: Croquet architecture

Large-scale deployment of collaborative Croquet spaces involves a combination of the peer-to-peer networking present in Croquet with a distributed mesh of Worldbase servers acting as metadata and object repositories and a series of persistent super-peers (Interactivity servers) providing persistence for locales within the virtual space [8]. The architecture is split into three functional parts:

1.) *Croquet peer-to-peer clients* running on users' personal computers. The clients software is responsible for rendering the virtual scene, and the Croquet peer-to-peer protocol keeps clients synchronized.

2.) *Interactivity* servers for persistent locales. Interactivity servers are responsible for maintaining the existence of a Croquet world even in the absence of any user's presence. Since Croquet is a peer-to-peer system, we make provisions for spaces to exist without any users (or peers) being present in the space.

3.) *Worldbase* repository servers. Worldbase servers are capable of storing serialized (flattened) objects representing 3D models, textures, pictures, hypertext links, and executable code. Worldbase servers act as a virtual library or digital commons for the system.

3. Collaborative authoring tools

Our research explores ways by which multiple authors can modify Croquet scenes and object by adding content, comments, and references. This is essential to support collaborative research and learning for scholars and students. By exchanging ideas and comments about content, users deepen their understanding and Croquet spaces act as a 3D wiki or blog with community-developed media, content, and commentary.

Annotations may be thought of as author-attributable content placed within a Croquet scene in association with (or reference to) a particular element of that scene. For example, an expert in archaeology may enter a Croquet scene that contains a 3D model of an archaeological artifact in need of some form of commentary. Using scene annotation capabilities, this user should be able to add commentary in association with specific aspect (or aspects) of the artifact. This new content would persist in the Croquet space and be made available to other users as a form of annotation attributed to the author. This new content should also be searchable, so that other users could locate all commentary on a specific artifact or commentary created by a specific author.

To this end, we have developed basic conventions by which annotations can be created independent of the form of media. We allow objects to become annotation even if the object was not originally designed to be an annotation. This is important so that we provide the richest possible set of tools for expressing comments and annotations. Our approach is to provide a way to dynamically add *annotation behavior* to an object after the object was created independent of the form the object takes (text, images, audio, video, 3D objects, or even entire 3D spaces). *Annotation behavior* means that an object responds to queries about what it refers to (the target artifact). So, any object can become an annotation that refers to another object. By optionally adding an *annotationOf* method to the basic rendering container in Croquet (the TFrame) we can turn any visible object into an annotation. The *annotationOf* method returns the name of the object that is the target of the annotation.

3.1 A design approach for scene annotation

A general annotation capability requires that we develop a framework for creating and marking objects as annotations of artifacts in a Croquet scene. It also implies support for selective viewing of certain classes of annotation (such as those created by a particular author). To address these needs, we provide a digital repository for storage and retrieval of annotations by a number of criteria. Once retrieved, it should be possible to selectively display and hide annotations based on a user's interest or viewing authorization. Given a common repository and appropriate meta-information associated with objects, it is possible to support search and history functions so that annotations may be selectively displayed. Here, meta-information consists of information that is needed for library/search functions. We do not require all object classes include meta-information required for search and retrieval (such as author name, key words, etc.). Instead, this meta-information is added at the time the object is stored to the shared object repository (the Worldbase). By including some form of timestamp or sequence number as meta-information, the evolution of a set of annotations over time can be visually explored.

Let us now look in detail at how we name, store, and connect annotations and objects together. We need a number of constructs to achieve our goals:

1.) A unique identifier for each object. Each object to be annotated in a Croquet scene needs a name by which it can be referenced. Unique IDs should be verifiable and automatically assigned without reference to a central naming authority in order to avoid creating a bottleneck in the architecture.

2.) *Tamper-evident, cacheable object/metadata containers (TCache objects).* TCache objects are used to package metadata along with a flattened representation of the object to which the metadata refers. The metadata is an open-ended list of attribute/value pairs (Table 1). The name (unique ID) of the TCache object is a SHA1 hash [9] over the attribute/value pairs as stored in a Croquet dictionary object. By comparing the SHA1 hash computed for a TCache object with its name, the TCache object can be verified as valid. We expect Croquet clients to aggressively cache TCache objects and share them via peer-to-peer distribution mechanisms, so some sort of tamper-evident storage mechanism is necessary to work with potentially un-trusted peers.

Table 1. TCache container.

attribute	value
object	serialized (flattened) object
author	author's name
creation date	date/time
optional info...	...
annotationOf	TCacheID of annotation target

3.) *The Worldbase; a shared repository into which TCache objects can be stored.* TCache objects need to be retrievable by their unique ID and searchable by attribute. We store both the artifact/object that is the target of the annotations and the annotations themselves in TCache objects in the repository so we can search and cross references them even if an object and annotations are used in more than one Croquet images/scene.

4.) *A convention that any Croquet object can be an annotation provided that it responds to an "annotationOf" message*. The *annotationOf* message returns the unique ID of the object to which the annotation refers (the target). Annotation objects may also position themselves in some spatial relationship to the target object of the annotation and render a pointer or connector to the target object. How annotations render themselves will vary, but at a minimum, annotations must be able to tell us what they refer to. When the target of an annotation or annotation itself is stored to local disk or placed in a repository, they are packaged in TCache containers. An annotation target's unique ID is stored in a meta-data attribute in the TCache so that we can search the repository and locate annotations that refer to a specified target.

To place an artifact or annotation in the Croquet digital commons, users publish the TCache container to the Worldbase. Because the annotationOf attribute is searchable in the Worldbase, anyone who wishes to find annotations for an artifact can query the Worldbase or their peers for TCache objects that refer to the given target. It is possible to store attributes such as annotation author, creation date, and other meta-information along with any annotation, so users can fetch collections of annotations by author, date, or other forms of metadata.

5.) *Ability to dynamically add annotation behavior to Croquet objects.* Not all objects can or should respond to the annotationOf message, and we want to allow for open-ended expansion of individual object's capabilities, so some sort of add-in behavior mechanism is desirable.

We address this issue by providing users with a way of dynamically adding (or removing) annotation behavior for an object. Right clicking on an object brings up a contextual menu where a number of behaviors (annotation, spin, drag, friction) can be added or removed from the object. We found that add-in behaviors are useful for more than just annotation, and added some motion behaviors so that an object could be positioned (using the drag behavior). Once the object is correctly positioned, the drag behavior can be removed so it is not accidentally displaced. This sort of add-in behavior mechanism makes it possible to dynamically control an object's capabilities on a per-object instance basis. By adding a TBehaviors to the basic rendering object in Croquet (the TFrame) users have control over a set of object behaviors without resorting to modifying the object's base class.

3.2 User interfaces for scene annotation

Our approach in developing user interfaces for scene annotation is to employ pull-down and contextual menus as a means of initiating the creation of annotations. These familiar user interface metaphors allow naïve users to quickly and easily access these functions. We are also exploring how familiar user interface conventions can be incorporated into the *filter portal* concept [10][11]. Filter portals can be thought of as looking glasses through which a different view or a user interaction mode may be displayed in the 3D scene. Our explorations in this area suggest that there should be at least two approaches to displaying annotations in Croquet. Annotations may be represented as visibly linked objects (Figure 2) or they may be a marker (displayed as a thumbtack denoting some hidden content that can be made visible when triggered by a user action - mouse over, click, etc.).

Annotations are created through the use of a contextual right-click menu on an object to be annotated. This presents users with options for the type of annotation to be created at the point of the mouse-click. Depending on the type of annotation desired, an appropriate set of authoring tools is presented to the user. By this convention, a user may rapidly develop text, audio, video or other forms of annotation of the scene and objects contained therein. Linking the annotation to the location of mouse-click by a visual connector makes it possible for users to create multiple annotations (perhaps even of different media types) to particular objects or parts of objects in a scene.

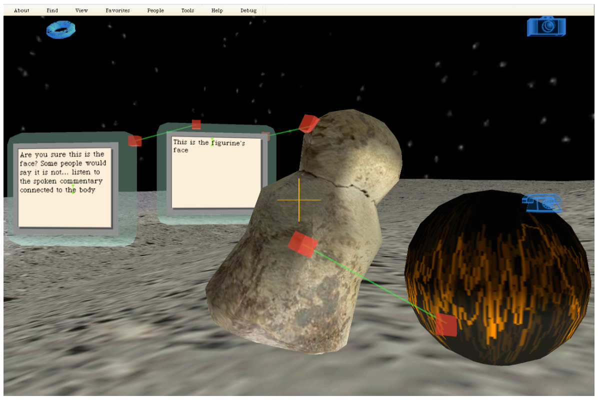

Figure 2. Stone artifact with linked text annotations (left) and audio annotation (right).

Annotations that are always visible can result in significant visual clutter. On the other hand, if an annotation is not visible by default, users can easily overlook it. This issue was the motivation for developing both thumbtack and visible link annotations. Thumbtack annotations are represented by pins on the object being annotated, and require users click on the thumbtack to see the annotation. Once clicked, a thumbtack is animated with a visible throbbing (growing and shrinking), to make it clear which marker is active. Further exploration of how to visually link markers on objects with annotations displayed in filter portals is underway. So that users can quickly switch modes between seeing the thumbtack marker and annotation, and seeing the unadorned object, the markers and annotations are only visible by looking through a filter portal (Figure 3). Visible links are implicitly rendered in the scene and are denoted by a green line between the object and its annotation (Figure 2).

Once the user selects the appropriate menu item for creating a new annotation, a window containing an authoring tool for creating that annotation is presented to the user, and is associated with the object by a visual connector. For text-based annotations the user is then able to enter annotation directly into the new window. For audio annotation, the user is presented with an appropriate interface for recording audio directly from their microphone or from a pre-recorded sound file. A similar approach can be taken for video. In these cases, the annotation is displayed as an audio or video object in Croquet. As such, it can be readily repositioned or edited by multiple users.

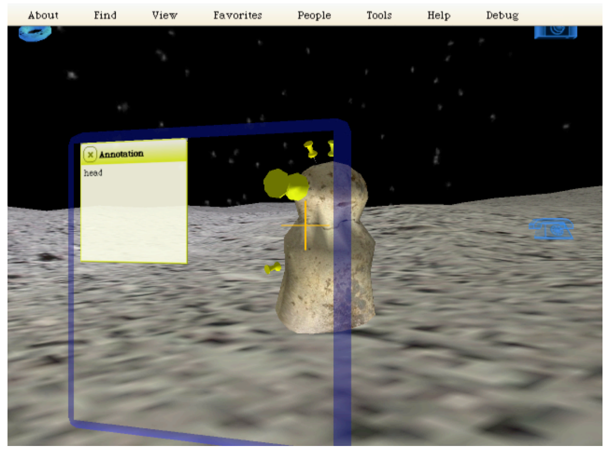

Figure 3. Thumbtack markers and annotations are only visible when viewed through a filter portal.

The annotation architecture makes it possible to annotate one object with another object so anything that can be represented within a Croquet space can also be used as an annotation. In our initial implementations, users click and drag with the middle mouse button from the model they wish to annotate to the object that will serve as the annotation. Upon release of the mouse, a visual indicator of the relationship (a green line) appears in the scene. Since all annotations are themselves user-created objects within a scene, annotations may be added to existing authors' annotations as well. In this way, users can comment on the comments of other users.

Annotations may also be easily detached from objects or deleted entirely. In order to delete an annotation entirely, the user can either choose Delete from the popup menu or delete the annotation object itself. To detach an annotation connection, one finds the small connection cubes either on the model or on the annotation and calls up the popup menu for the connection to select the detach option.

Because we want to allow objects and their annotations to be instanced in new contexts, we need a way to differentiate between the backdrop or unimportant parts of a scene and the important artifacts in the scene. For instance, in a virtual studio the ground plane, the sky, and a table may be thought of as furnishings that are not important, but a set of artifacts sitting on the table may be the focus of the work in the studio. We want to allow for users with different spaces to import artifacts and annotations without bringing the entire virtual studio of the author into their world, so we need a way to separate the studio furnishing from the important artifacts. The *behavior* contextual menu can also make it possible to delete the capability for an object to be annotated by removing this behavior. This allows users to distinguish between parts of the scene are considered important artifacts and the props and stage setting.

We should also consider the case where an *entire* space or world is to be the subject of study and comment. In this case, the environment as a whole is the artifact being studied. Since Croquet allows for nested worlds or spaces inside other worlds, we have several options for arranging these environments. We may place portals to the world(s) inside our virtual studio, or nest worlds inside each other. In either case, annotations can be placed both by the portal leading to the space and inside the space itself. Virtual paths and tours are an important way to view environments, with paths through possibly nested spaces becoming a form annotation and this is described below.

3.3 User-defined paths in VR spaces

Collaborative 3D spaces make it possible for authors to not only publish text, audio, video and objects directly within the simulation, but they also make it possible for these elements to be arranged in a particular order within the space. Thus, authorship of information in 3D spaces also includes the arrangement of particular chunks of information. In this way, courses of study can be literalized as paths through a 3D space. We have developed user interface tools that enable authoring of simple paths through Croquet-based 3D spaces by treating the path itself as a series of annotations. By arranging a series of linked annotations in the scene, users may define a path or course through a scene with commentary at each stop on the path. Other users may then identify selected annotations and traverse the annotation path so that a tour of a scene and its annotations that has been defined by one user can then followed by another user or groups of users. These paths might span nested spaces or portals or be confined to a single world.

3.3.1 User-defined beacons

Another approach to creating user-defined paths involves the creation of beacons. Beacons are markers or landmarks placed in 3D spaces by one author and then selectively viewed through a filter portal. Beacons provide a set of visual cues that can easily be followed as a user moves through an information landscape. A user is able to place beacons at particular locations via a right-click contextual menu that provides options for beacon placement within the user's view frustum. All beacons contain metadata about the beacon's author and certain conditions related to the beacon's placement. In this way another user employing a portal filter may selectively view the beacons that have been placed within the space by one or more other users or under one or more conditions. The distinction between beacons and annotations is that beacons represent a special case object that always represents a landmark. Linked annotations are a more flexible approach.

3.3.2 User-defined connectors

Another approach to creating user-defined paths within Croquet-based 3D spaces involves leveraging aspects of annotation functionality. Users can create multiple annotations that are linked by visual connectors (the same as those that connect an annotation to the object being annotated). These connectors each have a small box at both their origin and insertion. When a user clicks either end of the connector, a menu option provides the user with the capability to jump to the opposite end of the connector. This approach works to guide the user's viewpoint from one idea to another within the 3D space. For large 3D spaces, the use of this mechanism of translocation can be an effective means by which users are able to rapidly relocate along a predefined path through a vast and complicated information landscape. Using this method, an author of a path is able to define locations that are linked to one another via a connector that can be automatically followed by any user. Note that this approach accommodates branching paths since linked annotations may have many connections to other annotations. This is a precursor to other kinds of automated paths such as has been proposed for virtual museum applications [3].

4. User interfaces for import and storage

In our work we found it necessary to enhance Croquet so that we would have more options for importing and sharing objects. We provide several methods for users to import content into collaborative Croquet-based spaces. Basic import function for common file formats such as VRML, 3DS, MAX, DXF, and ASE are available and users can load files off their local disk. Locally loaded models are then propagated to other users in the same scene via Croquet's peer-to-peer network.

Creating a shared repository for the entire Croquet community is another approach to content import and sharing. Objects published to the Croquet Worldbase can be searched and instanced into Croquet spaces. This is important, since it allows objects to be shared between independent groups without requiring the groups to online concurrently or resident in a common space. The Worldbase repository acts as a digital commons for Croquet, and provides the capability to cross reference annotations that refer to a common artifact.

4.1 Access and placement of remote content

Users may fetch content from Croquet's Worldbase repository. Note that the TCache data format for storing objects to the Worldbase includes a globally unique naming convention so that independent research groups can create common references and annotations for objects without requiring that the research groups share the same Croquet space. Worldbase data objects also include metadata such as a unique identifier, time-to-live, and other attributes necessary for intelligent caching of objects. Anything placed or contained within the Worldbase has an attributed creator. In this way, Worldbase entries can only be written to through the action of authorized clients.

To retrieve an object previously stored in the Worldbase, a user selects an option from the pulldown menu. The user then enters the TCache ID for the desired object or some search criteria (creation date or author in our initial implementation). The results of the Worldbase query are inserted into the Croquet space (optionally along with annotations).

As users create objects and annotate them they may periodically publish their work. Right-clicking on any object allows the user to either *Save in the Worldbase* or *Save All Related*. Choosing to *Save All Related* will follow all the connections and store all related objects and their annotations. Each time the user saves to the Worldbase, a new version of the positions and relationships is created for use by the history mechanism. After a user has been working with an object - creating and editing annotations, saving periodically – they may want to view the different versions that they have created. Right-clicking on the object will allow them to *View History*. This provides users with a scrollbar to scrub back and forth through the history of the annotations and object versions for the states that they have saved.

4.2 An Object's Worldbase Information

Right-clicking on an object which has been stored in the Worldbase allows a user to choose *to View Worldbase Info*. Users can then see who created the object and when it was created. This is also how a user can view the object's TCache ID. The ID is useful for fetching the object and its annotations during a later Croquet session. Worldbase content is currently tagged with the author's identity, which defaults to the Squeak environment identity.

Annotations can be categorized based on the meta-information stored with them. This cataloging and categorizing can either take the form of SQL queries to the Worldbase storage outside of Croquet or as user-controllable views of annotation sets in Croquet.

5. Discussion

Our approach is to empower end users to actively participate in the creation, assembly, and architecture of applications they use. Consider the system in terms of Marshall McLuhan's idea of examining how a medium changes the user of that medium [12]. By supplying multiple content creation tools, we encourage users to create the content they desire. The information management tools allow the user to control their own environment, actively participating in information management and act as editors. We are creating a toolset in which users can actively participate in forming the dynamic architecture of their environment.

Croquet's collaborative environment allows direct manipulation of objects brought together from separate authors. The effect of multiple creation and editing mechanisms is multiplied by allowing the different application components to be used together. By providing a common set of tools for synchronizing and coordinating the shared virtual environment, Croquet simplifies the creation and sharing of interactive research/learning systems. This forms the basis for a cross-disciplinary simulation development and delivery platform that allows researchers to spend time on their unique content instead of reinventing the collaborative framework. By incorporating a digital repository infrastructure, Croquet spaces and artifacts can be queried, searched, and re-used in new contexts, further minimizing the cost of authoring content.

5.1 Tools with many uses

Arranging objects in the shared space provides an opportunity for expression by users as they organize the objects in the space. Connecting these objects with annotation connectors, and adding annotation text provides more opportunity for expression and is a 3D analog to blogs and wikis. These annotations can be stored and retrieved from the Worldbase. Users and groups of users can author and publish their individual resources within persistent 2D or 3D visualizations or simulations. They may build any number of private or shared “worlds”, making them accessible for others to explore via spatial portals.

By annotating artifacts with other artifacts it is possible for 3D collaborative spaces to serve as a medium for dynamic communication that moves users beyond the necessity for using text as the primary mode of communication. For instance, the history mechanism allows for playback of the evolution of a 3D concept map over time. Providing a scrollable timeline for the history of the evolution of the annotations can also act as a un-do feature. The user scrolls back to a desired state and then jumps in to create a new future. This also becomes a presentation tool, since users can scroll back to the beginning of time (when there are no annotations), and then scroll forward to gradually reveal content. Since annotations are objects, users could reveal not just text and graphics, but audio, video, portals to other worlds, simulations, and so on. This gradual reveal is one example of selective information hiding – a necessary technique for managing the complexity of community created content.

Many of these tools provide other uses or unifying elements. For example, we have used annotations as a way of finding new information and viewpoints. Annotations of annotations provide concept or idea mapping [13]. A history playback allows an exploration of the growth an annotation idea map in time. The next generation of these tools is being designed for even deeper reuse, with direct manipulation of more behaviors than the current drag, spin, friction, and annotation behaviors. We will continue in this line, in a manner similar to Self [14]. Ultimately, we should provide users with a rich palette of generalized behaviors that may be composed.

6. Conclusions

We have demonstrated how a variety of media can be used to comment on artifacts in a collaborative 3D environment, we have also demonstrated a mechanism where loosely coupled groups can independently publish annotations of artifacts, and the annotations from these groups can be viewed and explored. This automates some of the work of scholars, and facilitates more general communication between user communities, by building support for searching and joining together annotations stored in a common repository. This approach illustrates how significant synergies can be achieved when a virtual publishing system is designed for both academic and general use.

References

[1] J. Rodzvilla, Ed. We've Got Blog: How Weblogs Are Changing Our Culture. Perseus Press, 2002.
[2] B. Leuf, W. Cunningham. The Wiki Way: Quick Collaboration On the Web. Addison-Wesley, 2001.
[3] T. Barbieri, P. Paolini. “Cooperative Visits to WWW Museum Sites a Year Later: Evaluating the Effect.” Proc. of Museums and the Web2000, pp. 173-178, 2000.
[4] J. Gee. What Video Games Have to Teach Us About Learning and Literacy. Macmillan, 2003.
[5] R. Kalawsky. The Science of Virtual Reality and Virtual Environments. Addison-Wesley, 1993.
[6] D. Smith, A. Raab, D. Reed, A. Kay. “Croquet: A Menagerie of New User Interfaces.” Proc. of the Second Conf. on Creating, Connecting and Collaborating through Computing, pp. 4-11, IEEE CS, 2004.
[7] D. Ingalls, T. Kaehler, J. Maloney, S. Wallace, A. Kay. “Back to the Future: the Story of Squeak, a Practical Smalltalk Written in Itself.” Proc. of OOPSLA, pp. 318-326. ACM Press, 1997.
[8] M. McCahill, J. Lombardi. “Design for an Extensible Croquet-Based Framework to Deliver a Persistent, Unified, Massively Multi-User, and Self-Organizing Virtual Environment.” Proc. of the Second Conf. on Creating, Connecting and Collaborating through Computing, pp. 71-77, IEEE CS, 2004.
[9] The Secure Hash Algorithm (SHA-1), NIST, U. S. Department of Commerce, April 1995. http://www.itl.nist.gov/fipspubs/fip180-1.htm
[10] R. Kadobayashi. “P2P based Collaborative Annotation Environment for Digital Archives.” Proc. of the Multimedia, Distributed, Cooperative and Mobile Symposium (DICOMO2004), pp. 611-614, IPSJ, 2004 (in Japanese).
[11] D. Smith, A. Raab, Y. Ohshima, D. Reed, A. Kay. “Filters and Tasks in Croquet.” Proc. of the Third Conf. on Creating, Connecting and Collaborating through Computing, pp. 50-56, IEEE CS, 2005.
[12] M. McLuhan. The Medium is the Message. Signet. 1964.
[13] R. Abrams. “Meaningful Learning: A Collaborative Literature Review of Concept Mapping.” http://www2.ucsc.edu/mlrg/clr-conceptmapping.html
[14] R. Smith and D. Ungar. “Programming as an Experience: The Inspiration for Self.” Proc. of ECOOP '95. 1995.