Design for an Extensible Croquet-Based Framework to Deliver a Persistent, Unified, Massively Multi-User, and Self-Organizing Virtual Environment

Mark P. McCahill
Office of Information Technology
University of Minnesota-Twin Cities
mpm@umn.edu

Julian Lombardi
Division of Information Technology
The University of Wisconsin-Madison
jlombardi@wisc.edu

Abstract

We describe a design for a collaborative Virtual Learning Environment (VLE) to support massively multi-user and multi-institutional learning communities. This architecture extends Croquet to encompass a tier of distributed cache servers that allow intelligent caching and pre-fetching of objects, scene definitions, and textures. The design also provides for Interactivity servers tied to specific regions in the VLE. Interactivity servers provide persistence for neighborhoods, coordinate modification and transient phenomena for locales in the virtual space, and ensure trust relations by managing user identification, authentication, and authorization via integration with existing institutional authentication infrastructure. This system supports real-time interactions that promote a self-organizing, interdisciplinary knowledge sharing system within a scalable VLE in a manner that ensures accountability and trust relations.

1. Introduction

Current commercial e-learning platforms such as WebCT [1] and Blackboard [2] have given us "transactional environments" for education that provide a framework for delivery of course syllabi, presentations, text-based discussions and chat tools, and automated grading of quizzes and tests. These systems fail to provide support for deeper interaction and collaboration necessary to realizing a community-based, learner-centered "constructivist" pedagogy.

These and other course management and learning management systems (CMSs and LMSs) also fail to take full advantage of the powerful graphic and simulation capabilities of modern personal computers. Given these curent LMS/CMS systems, the communicative power of simulations and immersive social environments are difficult to achieve. Current CMSs and LMSs are designed primarily as centralized server architectures and have scalability problems that limit their use for large-scale cross-institution collaboration.

To move beyond the limits of the current generation of online learning systems in support of constructivist approaches to education requires a far more collaborative and interactive environment than is afforded by the web-page based tools of current CMSs and LMSs [3]. Because Croquet [4] combines powerful graphics and multi-user collaborations in a cross-platform environment designed for rich interaction and simulation, it provides an ideal basis for addressing many of the limitations of current learning systems. Although at present a promising technology, Croquet will require some extensions to be used on a large scale in higher education teaching and learning. This is because Croquet is not yet integrated with existing university identification, authentication, and authorization (IAA) infrastructure. Authentication and authorization are crucial for the implicit social contract present in real-world educational environments and its transfer to the online realm is a prerequisite to establishing trust relations needed for constructivist online learning.

Beyond providing for attributed identity, authorship, and ownership in the online realm, the Croquet architecture extensions we propose will enable smart caching and pre-fetching of objects. When large scale distributed online information systems first became popular with Internet Gopher [5] and then WWW [6], their architectural and scalability problems were not well understood. With the benefit of over a decade of hindsight, it is now possible to revisit the implicit architectural problems with popular distributed information systems and avoid some of the known shortcomings of current system architectures.

This paper proposes a number of extensions to Croquet that are now beginning to be developed as part of a multi-institutional effort to define an extensible Croquet-based framework for higher education. We expect that some of the details of what we propose will change as we progress, but the broad outlines should still be clear.

2. Learning from Other Architectures

Given today's dominance of the Web, it is easy to forget that providing backward compatibility to older information systems is vital during the launch and early adopter phase of new systems. To provide access to more content, Internet Gopher included gateways to existing ftp, Archie, WAIS, and USENET news systems. Web clients included Gopher support so that they could embrace and extend that system. Thus, it is vital that Croquet include a full-featured Web client, so that the legacy content of the Internet is accessible from inside the Croquet-based simulation. Since today's popular online learning systems present content via web pages, it is necessary to provide backward compatibility to provide the best of both worlds in the new system.

2.1 Distributed Authorship

It is axiomatic that we need to support many independent authors and domains of control to allow for exponential growth similar to what Gopher and WWW enjoyed. Unfortunately, this explosive growth came with a large hidden cost. Because there was no place in early versions of these systems to store meta-information, there was no standard way of identifying authorship, or that two documents were somehow related. Even after providing for explicit storage of meta-information [7], there was low user acceptance of the use of such meta-information since the manual tagging of documents and collections was generally regarded as not worth the effort.

Taking the path of least resistance, users published portions of their local disk's file system (Gopher) and hand coded HTML with ASCII text editors (WWW). The web browser wars of the 1990s fractured the nascent HTML standards, and vendors implicitly encouraged users to take advantage of varying web browsers' capabilities. Much effort went into maintaining browser compatibility rather than into building a better information system. The result of all of this is that the Web is for the most part a large unorganized mass of content of questionable overall quality. This situation is now being addressed in an *ad hoc* manner by using web pages as the presentation layer for proprietary authoring/content management systems. For Croquet to succeed in the higher education environment, the system should be able to automatically tag content with meta-information about authorship, content identity, and the like so that the entropy of the information mesh it creates can be minimized.

Since creating 3D content and scenes is expensive, establishment of a distributed repository where objects, textures, and scene components can be shared is important if Croquet is to become a compelling tool for educators. The repository we propose (the Worldbase) contains both content and meta-information that is automatically added to objects when they are placed into a scene and then ultimately promoted to the Worldbase repository. Since users are composing objects and scenes via Croquet, it is possible to sidestep the fact that users are loath to include meta-information with content by having the technology do it for them. Croquet in concert with the Worldbase should act as a content management and authoring system.

2.2 Caching Content

For speed, client application developers want to cache frequently accessed and slow to download content onto the user's local disk. Unfortunately, this is difficult for web clients given the lack of meta-information about the elements that make up a page. How does a web client reliably know that a graphic in the local cache is current and the same as one referred to on a page? It is telling that the *reload* button is a mandatory feature in web browsers. A reload function is necessary so the user can tell the client software it guessed wrong about caching images.

From another perspective, the value of server-side caching to minimize network costs or avoid congestion is seen in services such as Akamai [8]. These server caching services grew with the need to provision variable server capacity for special events that attract mobs of users. Server caching is also employed by network operators to trade local disk space for bandwidth to the Internet at large.

Architectural support for intelligent aggressive caching could help Croquet avoid the fate of most peer-to-peer systems in higher education. Because peer-to-peer systems employ caching and content sharing strategies that do not take into account relative bandwidth costs on and off the campus network, they consume excessive bandwidth at the gateway between campus networks and the commercial Internet. When campus network operators see what they perceive as excessive bandwidth bills, they react by scanning and banning, port blocking, or bandwidth throttling peer-to-peer services.

To address these issues, we propose that meta-information stored in the Worldbase would give Croquet-based clents enough caching hints to enable robust client, server-side, and peer-to-peer caching of content. This means carrying object attributes including a globally unique identifier, expiration time, author, and a digital signature to allow independent verification that the object has not been tampered with.

2.3 Search Engines

User maintained distributed information systems need an automated, searchable index or directory service that attempts to span the entire information space. This was seen with Archie (anonymous ftp), Veronica (Gopher), and AltaVista, Lycos, Google and a host of others for WWW. The lesson for Croquet is to make provisions for indexing and searching of the information space so that content can still be easily located as the system grows. Again, the ability to associate arbitrary metadata with objects should help with developing solutions to this problem that are unavailable in older systems. The 3D interface metaphor also carries the possibility of harvesting implicit information about relatedness of content based on physical proximity in the virtual scene. One can imagine an algorithm similar to Google's PageRank for categorizing scenes based on object inheritance and locality. Portals between scenes provide link-like information about relatedness, and looking for scenes that use instances of common objects should also provide more hints for the search engines.

2.4 Fragile Hypermedia

As pointed out by Ted Nelson [9] and others, the one-way links of the Web are extremely fragile. They often point to content that can move, change, or disappear. The result is a broken or dead link. The Web's links between objects provide much of the system's value, yet there is no real architectural support for automatically maintaining these links. Practical content management systems have been built that address this problem (such as Hyper-G/Hyperwave [10]). We are already proposing some content management system functions in Croquet to support tagging content with authorship and caching information. With this content management functionality, we also have the opportunity to tag hypermedia links in Croquet with two-way link information. By providing two-way links, the target of a link can notify those referring to the target when the target is to be moved, changed or removed.

3. Architecture Overview

Our goal is to build a massively multi-user virtual environment using the wisdom gained from previous efforts in distributed information systems. Users experience a direct view of a locale within a much larger and contiguous virtual environment (Figure 1).

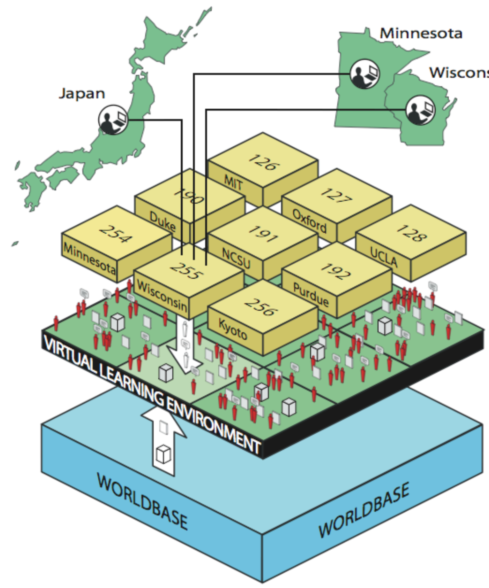

Figure 1: User experience/architecture

Users are able to move in a continuous environment from one locale to the next. Objects within the virtual environment function as hyperlinks to web pages and other Internet-deliverable services that can be viewed via the web browser embedded in a Croquet frame. Objects may also act as hyperlinks to other Croquet-delivered spaces or Squeak projects. Authenticated users of the system would be able to see and interact with each other while their client caches data about a locale and renders the scene in real-time.

Interactivity servers are essentially long-lived Croquet clients, and accept connections from all clients for a locale of the VLE to maintain transient or non-cacheable objects. Since the Interactivity server returns the list of users in the locale to new visitors, it is the natural point for authenticating a user's identity and providing session credentials.

The Worldbase servers distributes digitally signed, cacheable, authoritative data objects describing long term persistent aspects of any locale within the VLE and a network locator for each locale's Interactivity server.

We propose 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 layer of Interactivity servers. The Interactivity servers provide persistence for locales within the virtual space and integration with existing federated authentication infrastructures. Continuous persistence is important to allow for some sort of automated walking of the space as would be done by analogs of web crawlers employed by Google. These considerations lead the architecture to be split into three functional modules: 1) Croquet clients running on user's personal computers; 2) Interactivity servers maintained for locales that need long term persistence; and 3) Worldbase repository servers.

Worldbase servers function as a distributed cache system. We expect the Croquet clients to aggressively cache information fetched from the Worldbase servers, and to share the cached objects with other clients on their local network. A separate caching service allows network operators to explicitly trade local disk for bandwidth to the Internet at large. The Worldbase servers may be thought of as roughly analogous to Akamai servers on the web. A Croquet client's first priority is rendering the 3D scene, running simulation code, and keeping up with Croquet's TeaTime distributed object synchronization so there may be limited capacity (or bandwidth) for providing remote clients with objects from the client's local cache. The Worldbase repository augments the peer to peer caching capability of Croquet.

The Worldbase servers also maintain a mapping of which Interactivity servers are responsible for specific locales in the contiguous space. This mapping should be cached by Croquet clients. A client new to a virtual space needs information to rendezvous with other clients and find out who is active in the neighborhood. This rendezvous function is provided by the Interactivity server which makes it a natural point for integration with existing institutional authentication infrastructure. A client that cannot produce authentication and authorization credentials would either not be introduced to other clients in a restricted access locale or would only be provided with view-only access capabilities for the neighborhood. While there may be utility in providing public spaces that anyone can modify, it is also important to allow for spaces where only a restricted set of users have write or modify access. This restriction is again enforced by the Interactivity server and makes the Interactivity server a natural conduit through which objects can be promoted to the Worldbase repository. Since users would be authenticated before being given the capability to modify a scene, there would also be attribution of who modified the scene even in world-writeable regions of the space.

4. Worldbase Servers

Worldbase servers comprise a general purpose federated digital repository and and are capable of storing arbitrary bits representing objects including 3D models, textures, pictures, hypertext links, and code objects. Worldbase data objects will also include metadata such as a unique identifier, time-to-live, and other attributes necessary for intelligent pre-caching of objects by both Interactivity servers and clients. Anything placed or contained within the Worldbase will have an attributed creator. In this way, Worldbase entries can only be written to through the action of authorized clients.

Since users are expected to reuse or create instances of objects from the Worldbase at multiple locales within the VLE, the Worldbase server also carries an attribute that lists the locales that a given object is used in or referred to. This establishes two-way hypermedia links in the case where an object is a pointer to another location in the VLE. For objects that are links to resources in the Internet at large the list would allow for caching of resources held outside the Croquet world. For other object types, knowing where the object is used allows for compensation or acknowledgment of authors of popular objects and notification of affected locales if a problem is found in a cached object.

Objects stored on Worldbase servers will be made available to all clients and are essential to the construction of the relatively static elements in a scene as rendered by the client (Figure 2).

To illustrate how clients interact with Worldbase and Interactivity servers, consider the sequence of events as a user arrives in a locale:

1) Client queries a known Worldbase node for a user selected locale and requests updates to it's local data cache, Worldbase provides cache updates and the network location of that locale's Interactivity server.

2) Client establishes session with Interactivity server by presenting authentication credentials, requests transient locale data (rendezvous with other users in the scene) , and renders current scene.

3) Clients continuously update Interactivity server and each other with user's behaviors.

4) If the number of clients in a locale is large, the Interactivity server may provide all clients in locale with digest data informing clients of the relevant behavior of other users' representations in the current scene.

5) Interactive applications connecting via client protocols allow dynamic application output to be represented within the scene.

6) An authorized user or application may edit the locale data according to locale and Worldbase permissions, such edits are stored by the Interactivity server and may later be promoted to the Worldbase.

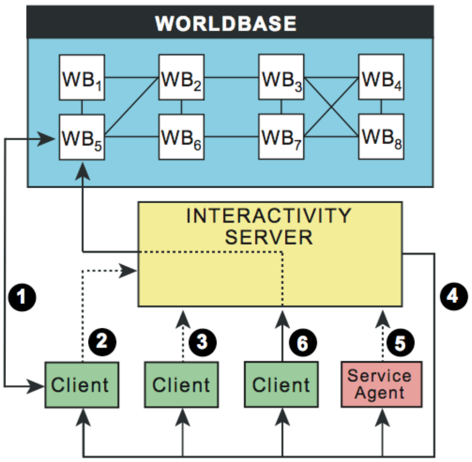

Figure 2: Client-server negotiation

Objects promoted to a Worldbase server will be replicated across the mesh of Worldbase servers on demand. For instance, a user might add a 3D protein model to a locale served by an Interactivity server in Minnesota with a long time-to-live value. The Interactivity server would then promote this model to a Worldbase server for propagation and caching, and update the quasi-static scene description and object list for this locale. The locale's object list and scene description are also promoted to the Worldbase. When a user in Wisconsin visits this locale, the Wisconsin client can ascertain if the quasi-static parts of the scene in the client cache are valid. If the client needs to refresh the local cache, it queries a Worldbase server in Wisconsin which fetches a copy of the information from the authoritative Worldbase server in Minnesota. Subsequent requests to the Wisconsin Worldbase can then be served from it's cache.

The notions of authoritative copies, time-to-live values and on-demand caching are intended to be analogs to how the Domain Name System functions. The Worldbase is where clients fetch the semi-static scene description and the addresses of Interactivity servers, so that the Interactivity servers can use their resources primarily to support dynamic behavior in a locale.

5. Caching

Maintaining authoritative time-to-live information on digitally signed cached objects allows the distributed Worldbase repository to function in concert with Croquet peer-to-peer sharing of cached objects. Clients can independently verify that the cached object is valid. If one client on a LAN has just fetched the set of changes for a locale, the other clients on the LAN are not forced to visit the Worldbase to fetch the same information. A side effect of longer time-to-live settings is that semi-static parts of a scene could not be changed immediately, and this behavior mirrors the real world. In the large contiguous virtual environment we envision, providing for landmarks and some stable structure is an aid to user navigation and orientation.

Adding an *embargo-until* attribute to cached objects provides for pre-fetching of objects that are not to be displayed until a specified TeaTime. This functionality improves the chances of saturating a user population with large objects in advance of user demand. For instance, to provide a virtual fireworks display at midnight on new years eve, pre-caching the fireworks objects by clients days in advance is prudent, so that the appearance of a flash mob does not swamp the system or available bandwidth. The caching system should allow for pre-shipping content to locations in the virtual space and specifying times in the future so that a thoughtful host can prepare for a party in advance. This functionality would need to be carefully authenticated and integrated with a capabilities-based security system such as "E" [11] so that this power is not used for unsavory purposes.

6. Interactivity Servers

To provide for a large contiguous space, we need to organize the objects in the space so that clients are not required to render the entire space any given time. By breaking the contiguous scene into adjoining locales, we can manage the rendering complexity by displaying adjacent locales at a low level of detail and not rendering distant locales at all. As a user travels within the large space they move between locales and are culled as an active player from locales that they are exiting.

Other functions of the Interactivity servers include caching of selected client-derived data. Objects authored on clients by users are authenticated and then cached on appropriate Interactivity servers prior to being promoted to the Worldbase. Thus, Interactivity servers mediate the promotion of information from clients to the Worldbase via a chain of trust architecture. They also deliver to clients the more transient phenomena within a particular locale such as who else is present in the neighborhood. Any particular scene rendered by a client is a composite of the more static Worldbase definition of the scene plus the transient phenomena (such as the behaviors and interaction with other users) that are tracked by the local Interactivity server. A particular Interactivity server works in conjunction with other Interactivity servers to segment the load of delivering a massively scalable common virtual environment (Figures 1 and 2).

7. Avatars and Crowds

The threat of large numbers of users simultaneously visiting a locale presents potential scaling problems both in rendering the avatars and in the amount of Croquet messaging traffic necessary to update avatar locations and each user's pointer manipulations. To make crowded scenes feasible, it may be necessary to display a crowd of users as a single aggregated visible avatar (a crowd represtentation) and cull the avatar messaging from the mass of users. In this scenario, the crowd avatar could be synthesized by the Interactivity server, and the server would provide the users entering the scene with a very limited list of other active players until the population density falls below the crowd threshold. Ideally, the limited number of users that clients render in a crowd scene would include people on the user's *buddy list* so they can find their friends in the crowd. The behavior sought here is one of soft failure so that as the system approaches its limits, performance gracefully degrades and users have a clue about why this is happening.

8. Authentication/Authorization

Robust authentication and authorization in a Croquet VLE is needed to support ethical collaboration, help to manage digital rights, and hold users accountable for their online actions. Since users will not be able to create fictive virtual personae in our integrative framework, communications between peers cannot be repudiated. With these security assurances in place, the virtual environment can provide trusted relationships necessary to collaborative learning and research. Members of the UM School of Engineering, for example, may be able to locate and leverage the online presence and content of colleagues within the Department of Computer Sciences at UW and securely initiate innovative cross-disciplinary collaborations that might not otherwise have emerged. The emergence of the Shibboleth [12] initiative makes it the obvious choice for cross-institutional authentication in the higher education community, so integrating Shibboleth with Croquet is a high priority.

9. Content Authoring

After considering the damage that hand edited HTML published without the benefit of any sort of content management system did to the Web, we are convinced that user content authoring for a Croquet VLE must be integrated into the Croquet client. By combining the client with a common shared repository and reasonable conventions for naming and referencing objects, we can build in robust content management and simplify object and model reuse for content creators. Given the cost of authoring 3D content, this is a vital consideration.

Providing for group collaboration and authoring to collect, discuss, organize and criticize knowledge is our goal for the Croquet VLE. In this form of authoring, being able to find and add content is crucial. Combining Croquet with the proposed Worldbase repository will make powerful searching and location tools possible. Through the Croquet VLE interface, objects within and beyond the current locale may be identified and located through the use of directories and searches of Worldbase entries. Once a target object is identified, users will be provided the option to be transported to the corresponding object within the VLE. During the load time of a selected resource, users will be moved in a continuous manner through the space. During such transit, users are provided cues about the overall knowledge structure, the situation and relationships of the target resource, and the existence of other users interacting within the same knowledge space.

Availability of simulation construction kits are another crucial part of authoring content for the VLE. Ideally, the user-level scripting capabilities planned for Croquet will help fill this need. Until there is a large quantity of compelling content available only in Croquet, it will also be necessary to address the need for backward compatibility with the Web.

When a web page is requested by either entering a URL or by clicking on a web object in the VLE, it is presented as a window within the scene. Because we seek to extend current VLE user interface metaphors while retaining backward compatibility, Croquet will require a better web browser than the current Squeak Scamper client. The shortest path to such a client is to make the Mozilla rendering engine a Squeak plugin. The intent is to integrate existing HTML page rendering technology into the virtual world rather than to build another HTML rendering engine. This strategy allows the client application to leverage present and future helper applications, authentication services, security, data access, parsing, and delivery capabilities contained within the current Internet application infrastructure, since such commonly available capabilities center around use of web browser applications.

In the VLE, content authoring may consist of collecting existing objects and adding commentary, creating simulations or toolkits for simulations, or presenting audio/video resources or web pages. Depending on a user's level of authorization in a given locale, they can define objects within the virtual environment that point to user-specified local or network resource data. Objects may also be published to the larger community depending on the user's identity and associated level of authorization for a particular region. When a user modifies objects within a locale, the Croquet client should pass the author's identity, object coordinates, an object identifier, any required assets to render the object to the locale's Interactivity server (and other clients currently in the locale). If the lifetime of the object is long, the Interactivity server would promote the object to a Worldbase server. Users who create such objects may also be permitted to remove them from the Worldbase. When removed from the Worldbase, objects would no longer be accessible by clients and, based on a time-to-live value, would eventually no longer be available for rendered within the VLE. In the event that others are using a child instance of a deprecated object, those users could be notified of its impending demise and take appropriate action.

10. Conclusions

To move beyond the current online learning environments for higher education implies a significant paradigm shift. Rather than limiting our vision to automating quiz grading and dispensing instructor powerpoint slides, we see Croquet as a first step toward a system designed for deep user collaboration, scalable realtime interaction, and authoring supported by a digital repository and an implicit content management system. Analyzing the successes and failures of the last 10 years in Internet distributed information systems, we propose a set of extensions to the Croquet architecture to support its use in higher education, and look forward to implementing this new paradigm.

11. Acknowledgements

We wish to thank Kevin Henninger and Peter Moore for their thoughtful comments and repeated reviews of the manuscript. We also wish to thank Preston Austin for stimulating and formative discussions of the concepts described herein.

12. References

[1] WebCT web site: http://www.webct.com/

[2] Blackboard web site: http://www.blackboard.com/

[3] J. Lombardi and M.P. McCahill, "Enabling Social Dimensions of Learning Through a Persistent, Unified, Massively Multi-User, and Self-Organizing Virtual Environment", *Proceedings of the Second Conference on Creating, Connecting and Collaborating through Computing (C5'04)*, IEEE Computer Society Press, 2004.

[4] D.A. Smith, A. Kay, A. Raab, D.P. Reed, "Croquet - A Collaboration System Architecture", *Proceedings of the First Conference on Creating, Connecting and Collaborating through Computing (C5'03) 0-7695-1975-X/03,* IEEE Computer Society Press, 2003, pp.2-10.

[5] M.P. McCahill, "The Internet Gopher: A Distributed Server Information System", *ConneXions*, Interop, July 1992, **6(7)** pp.10-14.

[6] T. Berners-Lee, R.Cailliau, J.F. Groff, B. Pollermann "World Wide Web: The Information Universe", *Electronic Networking: Research Applications and Policy*, **2**(1), pp.52-58.

[7] M.P. McCahill and F.X. Anklesaria, "Evolution of Internet Gopher", *Journal of Universal Computer Science*, Selected Proceeding of the Workshop on Distributed Multimedia Systems, IICM Graz, Austria Nov. 1994, pp.235-246

[8] Akamai Web site: http://www.akamai.com/

[9] T. Nelson, "Xanalogical Structure, Needed Now More than Ever: Parallel Documents, Deep Links to Content, Deep Versioning and Deep Re-Use", *ACM Computing Surveys* ACM Press New York, NY, USA 31(4), December 1999, article no. 33.

[10] F. Kappe "Introducing Hyper-G", *Hyperwave - The next Generation Web Solution*, H. Mauer, ed., Addison Wesley, 1996 pp. 44-54

[11] M. Miller and M. Stiegler, "E" Website: http://www.erights.org/

[12] Shibboleth Website: http://shibboleth.internet2.edu/