Croquet: A Collaboration System Architecture

David A. Smith, Alan Kay, Andreas Raab, David P. Reed

VPRI Technical Report TR-2003-002

Viewpoints Research Institute, 1025 Westwood Blvd 2nd flr, Los Angeles, CA 90024 t: (310) 208-0524

Croquet – A Collaboration System Architecture

David A. Smith
104 So. Tamilynn Cr.
Cary NC, 27513
davidasmith@
bellsouth.net

Alan Kay [1]
1209 Grand Central Ave
Glendale, CA 91201
alan.kay@
viewpointsresearch.org

Andreas Raab
University of
Magdeburg, Germany
andreas.raab@
squeakland.org

David P. Reed
MIT Media Laboratory
20 Ames Street
Room E15-492
Cambridge, MA 02139
dpreed@reed.com

ABSTRACT
Croquet [18] is a computer software architecture built from the ground up with a focus on deep collaboration between teams of users. It is a totally open, totally free, highly portable extension to the Squeak [5] programming system. Croquet is a complete development and delivery platform for doing real collaborative work. There is no distinction between the user environment and the development environment.

Croquet is focused on interactions inside of a 3D shared space that is used for context based collaboration, where each user can see all of the others and what their current focus is. This allows for an extremely compelling shared experience. A new collaboration architecture/protocol called TeaTime has been developed to enable this functionality. The rendering architecture is built on top of OpenGL [13].

KEYWORDS
Croquet, collaboration, User Interface, 3D graphics, Squeak, Smalltalk, TeaTime, OpenGL, peer-to-peer.

INTRODUCTION
Croquet was built to answer a simple question. If we were to create a new operating system and user interface knowing what we know today, how far could we go? What kinds of decisions would we make that we might have been unable to even consider 20 or 30 years ago, when the current operating systems were first created?

The landscape of possibilities has evolved tremendously in the last few years. Without a doubt, we can consider Moore's law and the Internet as the two primary forces that are colliding like tectonic plates to create an enormous mountain range of possibilities. Since every existing OS was created when the world around it was still quite flat, they were not designed to truly take advantage of the heights that we are now able to scale.

What is perhaps most remarkable about this particular question is that in answering it, we find that we are revisiting much of the work that was done in the early sixties and seventies that ultimately led to the current set of popular system architectures. One could say that in reality, this question was asked long ago, and the strength of the answer has successfully carried us for a quarter century. On the other hand, the current environments are really just the thin veneer over what even long ago were seriously outmoded approaches to development and design. Many of the really good fundamental ideas that people had were left on the cutting room floor.

That isn't to say that they thought of everything either. A great deal has happened in the last few decades that allows for some fundamentally new approaches that could not have been considered at the time.

We are making a number of assumptions:

- Hardware is fast – really fast, but other than for booting Windows or playing Quake no one cares – nor can they really use it. We want to take advantage of this power curve to enable a richer experience.
- 3D Graphics hardware is really, really fast and getting much faster. This is great for games, but we would like to unlock the potential of this technology to enhance the entire user experience.
- Late bound languages have experienced a renaissance in both functionality and performance. Extreme late-bound systems like LISP and Smalltalk have often been criticized as being too slow for many applications, especially those with stringent real-time demands. This is simply no longer the case, and as Croquet demonstrates, world-class performance is quite achievable on these platforms.
- Communication has become a central part of the computing experience, but it is still done through the narrowest of pipes, via email or letting someone know that they have just been converted into chunks in Quake. We want to create a true collaboration environment, where the computer is not just a world unto itself, but a meeting place for many people where ideas can be expressed, explored, and transferred.
- Code is just another media type, and should be just as portable between systems. Late binding and component architectures allow for a valuable encapsulation of behaviors that can be dynamically shared and exchanged.
- The system should act as a virtual machine on top of any platform. We are not creating just another application that runs on top of Windows or the

[1] Sr. Fellow Hewlett Packard and President, Viewpoints Research Institute, Inc.

Macintosh – we are creating a Croquet Machine that is highly portable and happens to run bit-identical on Windows, Macintosh, Linux, and ultimately on its own hardware… anywhere we have a CPU and a graphics processor. Once the virtual machine has been ported, everything else follows; even the bugs are the same. Most attempts at true multiplatform systems have turned out to be dangerous approximations (cf. Java) rather than the bit-identical "mathematically guaranteed" ports that are required.

- There are no boundaries in the system. We are creating an environment where anything can be created; everything can be modified, all while still inside the 3D world. There is no separate development environment, no user environment. It is all the same thing. We can even change and author the worlds in collaboration with others *inside them while they are operating* .

Figure 1. Croquet multi-user environment.

Croquet Is…

Croquet is a computer software architecture built from the ground up with a focus on deep collaboration between teams of users.

Croquet is a totally ad hoc multi-user network. It mirrors the current incarnation of the World Wide Web in many ways, in that any user has the ability to create and modify a "home world" and create links to any other such world. But in addition, any user, or group of users (assuming appropriate sharing privileges), can visit and work inside any other world on the net. Just as the World Wide Web has links between the web pages, Croquet allows fully dynamic connections between worlds via spatial portals. The key differences are that Croquet is a fully dynamic environment, everything is a collaborative object, and Croquet is fully modifiable at all times.

The current computer user paradigm is based upon a completely closed individually focused system. The user has a very low-bandwidth communication capability via e-mail, or instant messaging, but outside of some very simplistic document sharing capabilities, the user is quite alone on his desktop.

Croquet has been focused on high bandwidth collaboration from its inception. Simply put, the fundamental building block of the Croquet architecture is a system that makes every single object in the system collaborative.

Croquet's collaboration architecture is based upon the concept of replicated versioned objects coordinated by a universal timebase embedded in the communications protocol. This part of the architecture is referred to as TeaTime.

One way to think of the Croquet environment is as a high bandwidth conference phone call. Once a connection is made, the user not only has voice communication with the other participants, he also has the ability to exchange documents, collaboratively design systems, perform complex simulations, develop complex project plans, and manage complex projects.

Croquet utilizes OpenGL as the basis of its rendering /component framework called TeaPot. The architecture utilizes a semi-retained model, such that it uses a rendering hierarchy based upon dynamically composable objects, but each of these objects has full access to the OpenGL libraries and can extend the capabilities of the rendering engine in virtually unlimited ways.

Squeak is…

Croquet is built on top of Squeak [5], a modern variant of Smalltalk, hence it is a pure object oriented based system. This allows for significant flexibility for the design and the nature of the protocols and architectures that have been developed.

Squeak is a 21st century dynamic-object wide-spectrum operating and authoring environment derived from the 1970s Xerox PARC Smalltalk [4] system in which overlapping window GUIs, Desk Top Publishing, media authoring, and many other familiar software systems were first developed. Several of the authors of Squeak were principals at Xerox and were co-creators of many of the PARC inventions.

An essential part of our development process is Squeak's ability to keep the system running while testing and especially while making changes. Squeak allows even major changes to be performed incrementally and they take no more than a fraction of a second to effect. Another key feature of Squeak is its generalized storage allocator and garbage collector that is not only efficient in real-time (so that animations and dynamic media of many kinds can be played while the gc is collecting), but that allows reshaping of objects to be done safely.

Related Work

There are a number of seminal efforts over many years to which Croquet owes a great deal. Many of these early efforts were the first building blocks of the current popular windowing computer interface and usability paradigms. What is particularly interesting is that the focus of the Croquet project tends to be on the parts of these early efforts that were not picked up by what has become mainstream computing.

Sutherland's work [20] on direct manipulation and modeling of graphical object based entities clearly established the first true fundamental steps toward an interactive human computer user experience. Not only did he establish a great deal of the fundamental methods for how to create and manipulate interactive environments that are still quite relevant, but his focus was on creating a tool that would fundamentally amplify human capabilities.

His further work on the "Ultimate Display"[21] – the first immersive 3D experience, demonstrated the potential, still unrealized, of 3D interactive environments as the basis of a complete user experience to display and interact with computer data, creating "a looking glass" into what he described as a "mathematical wonderland." His vision of the system would represent data in 3-dimensional form, allowing the construction of entirely believable 3-dimensional, computer controlled, virtual worlds. He went much further than this in his description of the potential. "The ultimate display" he wrote, " would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such room would be fatal. With appropriate programming such a display could literally be the Wonderland into which Alice walked."

The efforts at Xerox PARC under the leadership of Alan Kay that drove the development of both pure object oriented development environments in the form of Smalltalk and powerful bit-mapped display based user interfaces was key [7]. In some ways, all we are doing here is extending this model to 3D and adding a new robust object collaboration model.

Douglas Engelbart's videoed first demonstration in 1968 of everything from a mouse to hypertext, object addressing and dynamic file linking, and especially shared-screen collaboration involving two persons at different sites communicating over a network with audio and video interface has been a major inspiration to this project. It is telling that this level of rich, deep collaboration between widely separated physical environments has still not been properly achieved. [2]

The Croquet component model architecture is similar to the OpenDoc system developed by Apple [1] and the Squeak 2D Morphic architecture developed by John Maloney [8] Both of these were designed around composable 2D objects – or components. The main ideas behind these systems are that the majority of the environment interactions that the components have to deal with are already available in the base classes that make up the system. The programmer's task is simply to override the behaviors of the objects – how they render themselves, and how they respond to "stimuli" from their surrounding components and from the user. Then the programmer and ultimate user can "compose" these intelligent blocks to form a useful document or application.

Smith's work on ICE – the Interactive Collaboration Environment, a multi-user shared component environment and later the Virtus OpenSpace architecture [17] acted as an important guide to the resulting Croquet system and in a sense Croquet is a far more complete result of this work.

Fisher et al [3] developed a powerful, totally immersive 3D working environment. This system included the ability to dynamically interact with the system via 3D menus and window documents, and the ability of the user to directly manipulate his position and orientation inside the world and interact with the objects that inhabited it. Further, the system could interact with the user as if he were just another object inhabiting the space. The best example of this was the virtual escalator that the user could step on that would then carry him up to another floor.

The TeaTime time based collaboration protocol/architecture is directly based upon early work of Reed [14, 15]. Jefferson's work [6] is related, but does not include the idea of maintaining a partial history, managing replicated objects, or incorporating two-phase commit. Miller and Dennis's Timewarp [9] protocol extends Jefferson's work to support "multiple versions", a central concept in TeaTime. Mirtich [11] employed the timewarp model to implement a graphics simulation architecture for maintaining complex physical modeled state.

CROQUET ARCHITECTURE

Like any complex system, it is impossible to account for the architecture of Croquet without describing its various pieces and the interrelationships they share with each other.

Croquet has been designed to operate as a peer-to-peer architecture. This ensures the greatest level of flexibility in the design of the system and its ultimate usability. All objects are symmetric in their ability to act both as local and remote recipients of the same messages. There is no intermediate step to be performed to interpret and rebroadcast these messages. A central server can be used to establish the initial connections.

The peer-to-peer architecture is also in keeping with the philosophical roots of Croquet, which is to act as a broad-band phone call (or better, as a conference call). Though a central server may play some role in a phone connection, its role is strictly limited to redirecting the pertinent information with no changes. Any messages sent from a