Crossjam: A Language for Authoring Hypermedia Scripts
Qualifying Exam Proposal: Executive Summary

Brian M. Dennis *
University of California, Berkeley †

Abstract

Hypermedia authoring systems are becoming a crucial technology in the computer science industry. We have the infrastructure and incentive to deliver digital media. The industry is awaiting authoring tools that skillfully allow for combining these media along with interaction to form hypermedia documents and applications.

Authoring can be broken down into three major areas: structural organization, interactive behavior, and graphical design. No current authoring system simultaneously supports clean specification of each of these elements. I propose Crossjam as a language for constructing scripts for hypermedia applications. Scripts are complete specifications of hypermedia applications defining structure, behavior, and presentation. The design of a script language is the focus of the dissertation.

This work is novel in that Crossjam is the first language to support formal specification of all three aspects of hypermedia authoring. Beyond designing and implementing the language, I will also develop a hypermedia authoring environment to demonstrate how a clean, complete language specification can enable advanced authoring tools. I will also demonstrate how Crossjam supports new authoring metaphors, interactive techniques, and hypermedia construction mechanisms. The included research plan outlines a course of development and implementation that is reasonable and timely.

*xjam@cs.berkeley.edu

†This research has been sponsored in part by the Advanced Research Projects Agency (ARPA) under Grant MDA972-92-J-1028, by the NSF under grant CDA-8722788, and an IBM Fellowship. The content of this paper does not necessarily reflect the position or the policy of the United States Government or IBM.

1 Introduction

Hypermedia authoring systems have become crucial to the advancement of the computer science industry. The growing infrastructure and economic incentive for delivery of digital text, video, sound, images, and animations now await tools that can effectively combine these media. Authors are muddling along with the current set of tools, but few if any are satisfied with their development environments. The multimedia industry awaits tools that can truly assist them in quickly developing hypermedia documents and applications.

Current hypermedia documents and applications are mainly built in one of three ways. A common means to developing such documents is to use the World Wide Web's language, HTML, to implement the document. Initially designed for the construction of distributed hypertext, HTML and the World Wide Web have been coopted by many as a hypermedia delivery platform. While HTML has fine facilities for describing the structure of a hypermedia document it fails to provide adequate mechanism to specify the graphic design of a hypermedia document, an essential element of expression to digital artists. This is one reason why we see such efforts as Netscape's HTML extensions and Adobe's attempt to incorporate PDF into the World Wide Web framework. Also, HTML is a fixed specification for expressing the structure of documents. Logical organizations outside the class of documents expressible by HTML do not fit into the World Wide Web framework. Since, the HTML specification is only slightly extensible, this are no provisions for custom structures.

Another means to constructing hypermedia documents and applications is to use a commercial authoring package such as Apple's HyperCard, Macromind's Director, or Authorware's IconAuthor. All of these applications provide productive development environments that even non-programmers can access. Unfortunately, all of the commercial offerings force the author into a rigid metaphor that the author has to mold his problem to. For example, there is no way to augment or alter the card/background/stack metaphor in HyperCard. This prevents authors from imposing their own logical organizations on the hypermedia application.

A third means to building hypermedia applications, is to use a general purpose programming language with extensions for multimedia content. Many multimedia designers are now taking the path of combining Visual Basic with multimedia VBXes to build hypermedia applications. This allows for increased flexibility for the programmer and can lead to better performance in the application. This tack though forsakes the favorable authoring environments of the commercial systems. There is also the need to build up the considerable infrastructure that such systems provide including access to graphical user interfaces, digital media presentation, and advanced presentation capabilities. Finally, this model tends to lead to a custom solution for every hypermedia application and discourage reuse of previous solutions.

To get to the root of the problem we must break authoring into three distinct areas. First, every hypermedia document has a logical organization that helps the author to manage the content and linking. This organization or structure has to be designed by the author. Second, the author has to develop the means by which the user can interact with their hypermedia application. Specifying how the application handles user gestures is a major part of the author's task. Finally, hypermedia authors design the presentation of the hypermedia content. This includes the graphical display of static digital elements and the temporal presentation of time based media. *No hypermedia authoring system today simultaneously supports a formal specification mechanism for all three areas of the hypermedia design process.*

Clearly since hypermedia titles are being delivered today, there are means for designing in all three arenas. A unified mechanism for formal specification of hypermedia documents provides a number of advantages over current technology though. Structure, behavior, and presentation all depend on and interact with each other. Attempting to separate them only limits authoring capabilities by preventing synergy between the elements. A unified specification mechanism is necessary. Formal specification allows for compact expression and precision of design. A well designed formal specification mechanism, e.g. language, enables clean, sophisticated synthesis and analysis of the hypermedia design, by humans or under program control. Finally, a language influenced by hypermedia authoring concerns can reduce development time by making common idioms convenient to use.

To the end of supporting hypermedia authors' requirements to specify struture, behavior, and presentation, I propose the design and implementation of a language entitled Crossjam. Crossjam is novel in that it will incorporate idioms to handle formal specification of the structural, interactive, and display aspects of hypermedia documents and applications within one language. This language will subsume current hypermedia scripting language capabilities and demonstrate new metaphors for constructing hypermedia documents. A significant portion of the research effort will be spent on implementing a hypermedia authoring environment centered around building Crossjam scripts. The goal of this research is to demonstrate that a clean, well designed means of formal specification can lead to authoring systems with capabilities beyond what we have today. Capabilities that can enable new and different types of hypermedia documents.

1.1 Overview

The remainder of this proposal is organized as follows. First, I attempt to further motivate the need for this research by demonstrating real world problems within current hypermedia authoring systems. I further attempt to motivate this research by hypothesizing some advanced hypermedia applications that are beyond today's hypermedia authoring tools. I then move on to discuss related research in this field. From there, I discuss the preliminary design of Crossjam and an architecture in which implementation of the language can be completed. This discussion is followed by a research plan covering analysis of the required infrastructure, outlining implementation goals, presenting a timetable of milestones. Finally, I conclude with a summary of the proposal and the research goals of the project.