Pachyderm Request for Partnership

(Sent to the NMC 11/10/2002)

Seeking Partners

The San Francisco Museum of Modern Art Interactive Educational Technologies group is seeking university technology partners to continue the development of Pachyderm, our multimedia authoring and publishing tool. We believe that with additional development the tool could be made available to the higher education and museum communities interested in online learning and multimedia production.

Pachyderm Overview

The San Francisco Museum of Modern Art Interactive Educational Technologies group has developed a multimedia authoring and publishing tool called Pachyderm. It was created to allow education staff at SFMOMA to quickly author interactive multimedia presentations. Pachyderm enables us to publish Flash screens that draw data from a database, without having Flash or SQL professionals on staff. It allows us to author presentations of varying size and complexity, and to publish the same presentation for the web, kiosks, or for a fixed disk such as a CD ROM or DVD.

Pachyderm was designed so that we could publish and continually update our flagship multimedia program *Making Sense of Modern Art*. We have also used it to create a variety of other presentations, modifying the published data or the Flash templates to suit the needs of a particular project. To date, the tool has been used for the following online features:

Making Sense of Modern Art	www.sfmoma.org/msoma
Making Sense of Modern Art: Gerhard Richter	www.sfmoma.org/richter
Eva Hesse	www.sfmoma.org/hesse
Ansel Adams at 100	www.sfmoma.org/adams
Art as Experiment, Art as Experience	www.sfmoma.org/anderson
Koret Visitor Education Center	www.sfmoma.org/education/kvec/
Whitney Biennial	www.whitney.org/2002biennial/

Documentation about the tool itself is available at: test.sfmoma.org/demos/pachyderm/index.htm

Moving Forward

While Pachyderm's original purpose was to publish *Making Sense of Modern Art,* our experiments with expanding its use have been so successful that we would like to create a version 2.0 to improve its functionality and make it available to the higher education and museum communities. We are seeking partners to work collaboratively to develop Pachyderm 2.0 and ensure its stability.

The following Action List gives some background on the areas of work that would need to be addressed. The Action List is provided as a starting point for establishing technical specifications for version 2.0. We are open to exploring different ways of structuring the development process and specifications with partners.

A note on the terms: "Pachyderm" describes a suite of tools, which will be referred to as The Import Tool, The Authoring Tool, The Publishing Tool, and the Pachyderm Database.

Action List:

I. Database

Modify Import Tool to receive XML data

Modify SQL database to permit input of object data

Modify SQL database to handle XML data

II. Interface

Add screen attribute for grouping screens by project

Add export function to Administration Tools

Standardize terms in onscreen instructions and field names

Standardize interface page formatting

Add user-level control of resources:

- **Shells**
- **Data sets**
- **Asset sets**
- **Object Label styles**

Expand interface to permit input of object data

III. Authoring / Publishing Tool

Modify Authoring Tool and Publishing Tool to permit linking between all screen types

Modify Publishing Tool to automatically generate index pages

Improve flexibility of installation and configuration options

Modify Publishing Tool to publish XML data

I. Database

Revisiting the Data Import Process

Action Item:
Modify Import Tool to receive XML data
Modify SQL database to permit input of object data

The present workflow model involves two databases: the MSOMA database is built in FileMaker, and is served on a local network to allow multiple users to view or input data related to museum objects (objects, hereafter), assets, rights tracking, and to some extent, content. A large subset of this data is imported periodically into the Pachyderm database, built in MS SQL. The Pachyderm database contains presentation data, entered by users via the Authoring Tool's browser-based interface. In effect, users must input data into two different databases that serve different, but related, functions.

It is generally agreed that maintaining two databases for the purposes of the publishing tool is impractical, but the ideal configuration is not perfectly obvious. Many institutions have a collections management system (CMS) that contains their object-related data, and would import this data into Pachyderm using the Import Tool. Theoretically, the Import Tool could be mapped to any CMS, but a more practical approach might be to select a single XML schema that can be imported into Pachyderm. Each institution's CMS would then be configured to export data to match that schema.

For an institution that does not maintain a CMS, it seems most logical to enhance the Authoring Tool's browser-based interface so as to allow users to enter all data into the Pachyderm database, and Pachyderm would then be the primary repository of object data, asset data, and presentation data.

Another solution might be to combine both approaches, so that users of Pachyderm could choose to input all data directly, import data from another database, or a combination of the two.

Adhering to Standards

Action Item:
Modify database to handle XML data

IET is currently investigating different standards for structuring presentation data, which could be applied to Pachyderm's input and output methods. The two main purposes are:
- to allow structured content to be imported into Pachyderm, and
- to allow greater potential for re-usability of content authored with Pachyderm

In the current configuration, the screen data is segregated into separate tables according to screen type, and each screen type has a set of fields unique to its type. In our move toward using standard data formats, the practice of storing data based on presentation styles is called into question. To maximize the reusability of the authored content, field names could be genericized so as to reflect only heirarchical and functional relationships, not the superficial characteristics of their presentation format.

Among the formats under consideration are: EAD, METS, and MPEG7.

II. User Interface

Improving Data Management

Action Items:
Add screen attribute for grouping screens by project
Add export function to Administration Tools

Presentations authored in Pachyderm may be intended for a single publish or for iterative publishing. If a presentation is intended to be published iteratively, it may make sense to store the presentation data within Pachyderm at all times. However, if a presentation is created for a temporary use, there is no reason to store its presentation data with the more permanent content. Currently, there is no simple way of isolating the content of one presentation from that of another.

The proposed solution is to assign each screen to an individual project or type so that users can more easily navigate through long lists of screens, either by choosing to view only a subset or by actually removing the outdated material from the database. Data sets of a more temporary nature could be purged after first being saved in a format that could be re-imported when needed. This would require a basic export function built into the interface. (The hierarchies recognized by the publishing component as described in appendix B also need to be recognized here -- If a presentation is exported, all of its dependent screens should be exported along with it)

Working Toward a Self-evident Interface

Action Items:
Standardize terms in onscreen instructions and field names
Standardize interface page formatting

The people who use the Pachyderm tool are writers, researchers, interns, et cetera, and while some have backgrounds in multimedia, it is important that the interface appear simple to people with only moderate technical experience.

During Pachyderm's development process, the tool's user interface was discussed relatively little. In some cases the result is straightforward and adequate, but there are numerous inconsistencies and gaps that give rise to error. There are

- discrepancies of terminology in onscreen instructions
- Buttons or controls which are hidden or hard to find
- No single standard for the format of the pages
- Fields which are not clearly identified

A number of these issues could be remedied by revisions to html pages, while others may require changes in asp or the database. Designs for the revised pages can be provided by the IET group, which has used the tool in its current state continuously for two years, then implemented in the server-side code by the developer, and further tested in usability studies with subjects unfamiliar with the tool.

User-level Management Tools

Action Items:
Add user-level control of resources (shells, data sets, asset sets, object label styles)
Expand interface to permit input of object data

Through authoring numerous presentations of varied scale and style, we have proven Pachyderm to be a versatile and flexible tool. However, when implementing these variations, much of the modification takes place after the presentation has been compiled by the Publishing Tool. Shells are modified using Flash, data files are modified using text editors, and the files modified are manually placed into the directory structure, overwriting those placed there by the publishing tool.

Although adding files within the directory structure is technically simple, it is potentially time-consuming and error-prone. Furthermore, certain changes require files to be moved or replaced on the Pachyderm server -- this is also risky and raises some user access issues.

Ideally, all of the modifications should happen within the authoring environment – no modifications made post-publish and no changes in server configuration. These controls would include:

- choosing a set of shells, or Flash templates
- selecting a set of media assets
- selecting a style for object labels for a presentation (e.g., full museum style or abbreviated academic)

Customized resources should be loaded onto the Pachyderm server, where they become available within the authoring environment. The user could leave defaults in place or select the resources they had uploaded.

The current configuration of the tool also requires object data to be imported from another database (as described in *Revisiting the Data Import Process*), which, although providing certain advantages, is cumbersome when creating simple presentations. If the authoring environment were expanded to permit direct user input into the object data fields, certain operations could be performed much more quickly. This could be implemented as a series of simple input forms, accessible as part of the authoring environment.

III. Authoring / Publishing Tool

Free-form Architecture

Action Items:
Modify Authoring Tool and Publishing Tool to permit linking between all screen types

The way in which the screens created in Pachyderm may be connected to one another is limited by design. Heirarchical structures that were envisioned for MSOMA were the blueprint for the tool's capabilities. Recognizing that these structures are one "style" of content delivery and are by no means universal conventions, we hope to expand on these capabilities toward more universal functionality.

Because the publishing tool is built to recognize certain hierarchies, it publishes "pyramids" of linked screens. In the original model the types of screens that one could link to from any given screen were highly restricted — each tier in the hierarchy would consist entirely of one class of screen. Later, some of these restrictions were removed, allowing different classes to occupy different tiers. However, the loosening of restrictions has not been applied to all aspects of the authoring and publishing tools.

Creating an index for a group of Screens

Action Item:
Modify Publishing Tool to automatically generate index pages

Whether used as a temporary placeholder for a custom element, or as a functional site index, we may use the "index" shell to connect a group of screens. Currently the publishing tool does not support this shell, therefore its data file must be hand-coded.

To automate the creation of this .ini file would be an extremely useful addition to the Presentation Editor and the publishing tool. The user would select an option in the Presentation Editor to "include index," which would cause all the other screens selected in the presentation to appear on the index page. This way, users can quickly publish a group of screens and easily navigate between them, even if they are authored without any links to each other.

Installation and server configuration

Action Item:
Improve flexibility of installation and configuration options

To move toward a system which is easily installed in different environments, the Tool needs to have some flexibility of configuration. At SFMOMA its components are installed on a single Windows 2000 server running IIS and MS SQL. Some institutions may choose to separate the media assets and other content from the application, or host the components on different servers, depending on the other systems they may already have running.

In anticipation of these requirements, file paths should not be hard-coded, but instead must be customizable to accommodate assorted configurations.

We also intend to explore the ramifications of moving the tool to an open-source environment, a change which could considerably broaden Pachyderm's potential user base.