



Lecture 2 – CollectionSpace intro

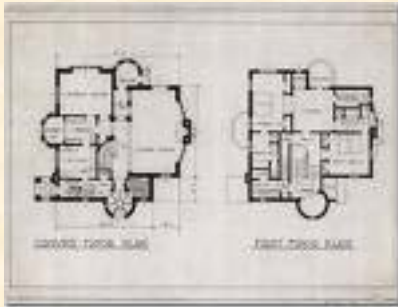
i290-rmm

Patrick Schmitz



UCB Context: The Problem

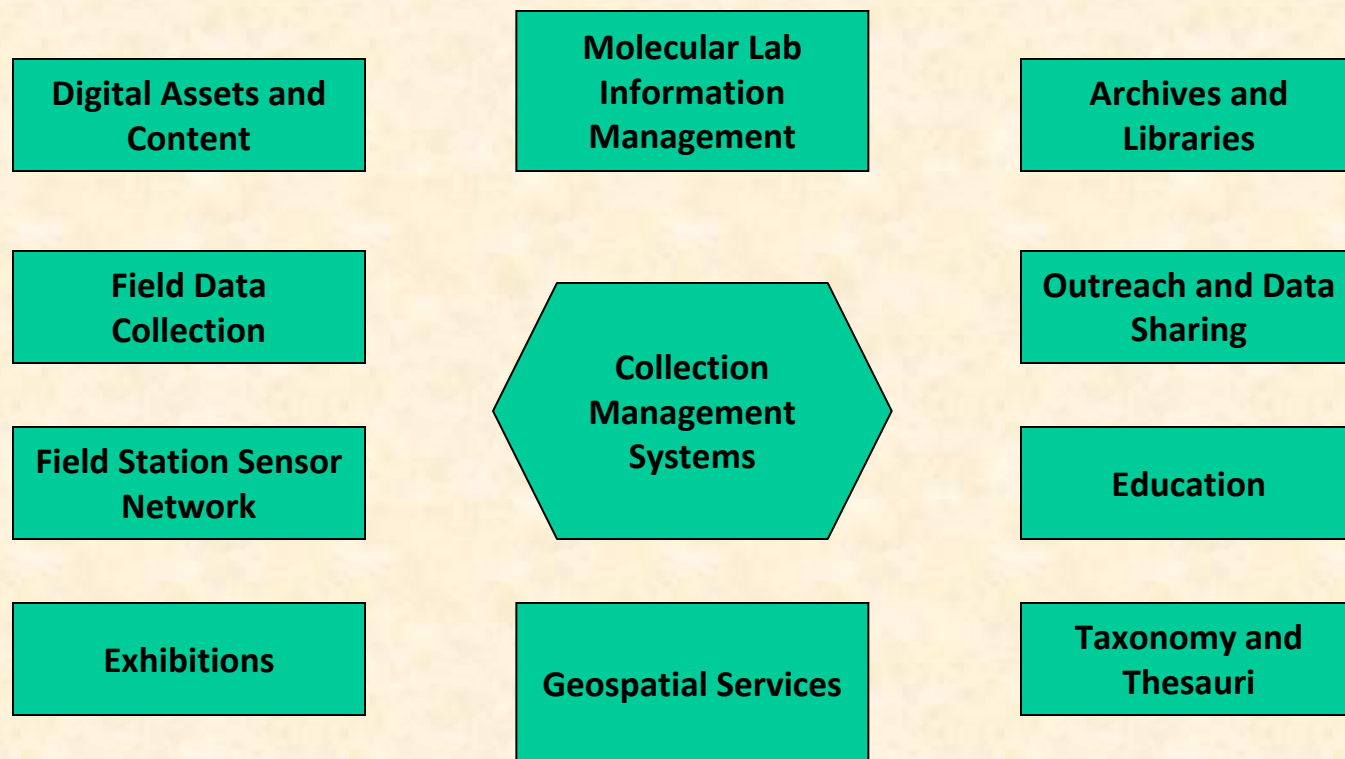
UC Berkeley Collection Management Systems

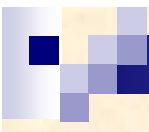


- Berkeley Language Center's Archival Catalog & Circulation System (Berkeley Language Center)
- CineFiles (Pacific Film Archives)
- SAGE (UC Botanical Garden)
- History of Art Visual Resource Collection (HAVRC) (Department of History of Art)
- Specimen Management System for California Herbaria (SMASCH) (University & Jepson Herbaria)
- Slide & Photograph Image Retrieval Online (SPIRO) (Architecture Visual Resources Library)
- PAHMA Collections (BNHM Consortium, Phoebe A. Hearst Museum of Anthropology)
- Biocode Specimen Database (BNHM Consortium)
- Essig Specimen Database (BNHM Consortium, Essig)
- HERC Specimen Database (BNHM Consortium, HERC)
- UCMP Specimen Database (BNHM Consortium, UC Museum of Paleontology)
- MVZ/Arctos Specimen Database (BNHM Consortium, MVZ)
- Plus ... Bancroft Special Collections and many others



Collection Management Systems – the center of scholarly ecosystem





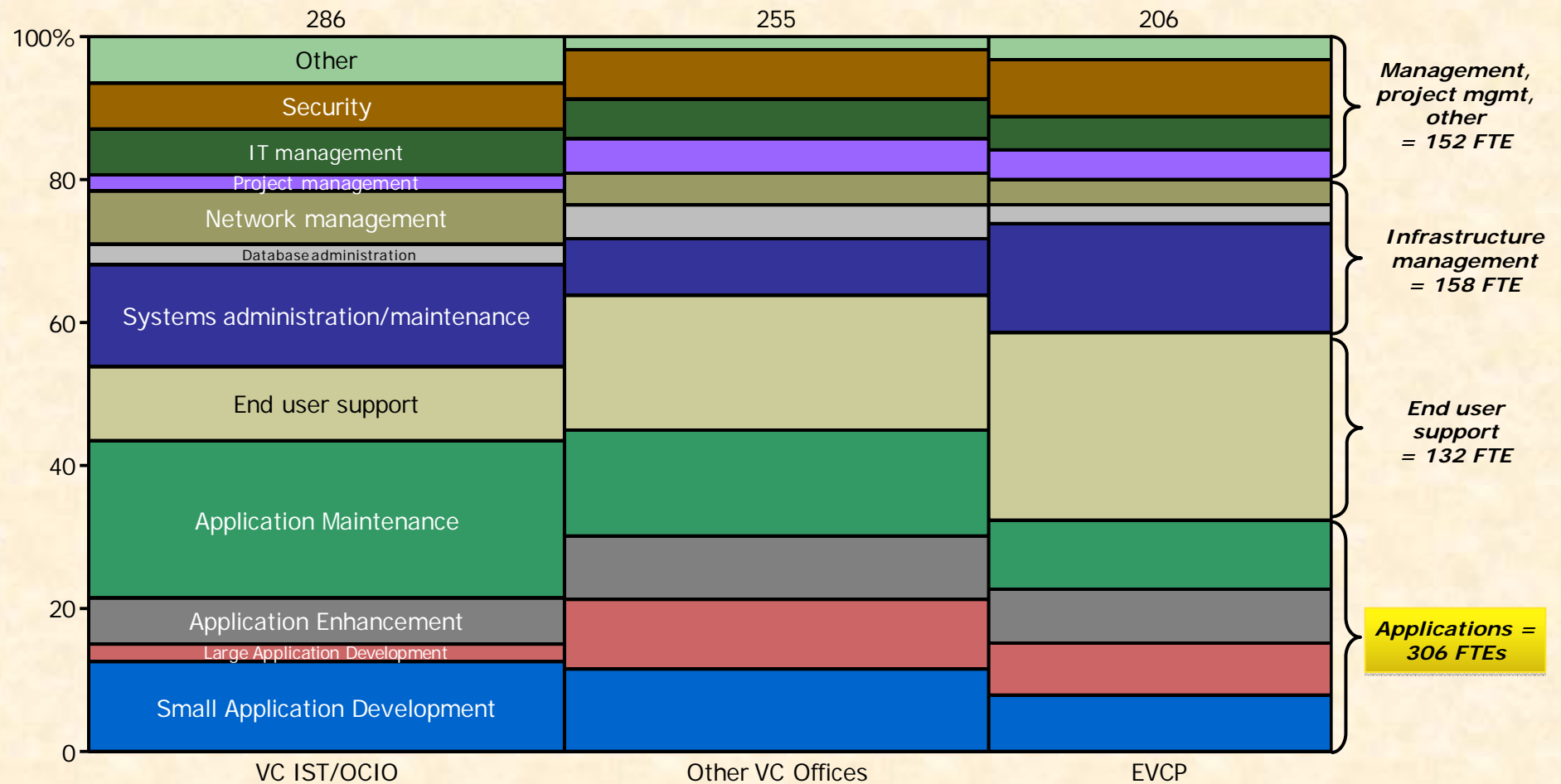
The Last 25 Years

- Too many systems, too many technologies
 - Millions of objects, artifacts, specimens
 - Managed in at least 20 very different collection management systems
 - Running on about 15 hardware platforms
 - Maintained by about 10 different technology groups, from amateurs to professionals
- Aging legacy systems
- Insufficient and inadequate funding models
- Unclear governance and decision-making

UC Berkeley currently spends \$125M+ on Information Technology

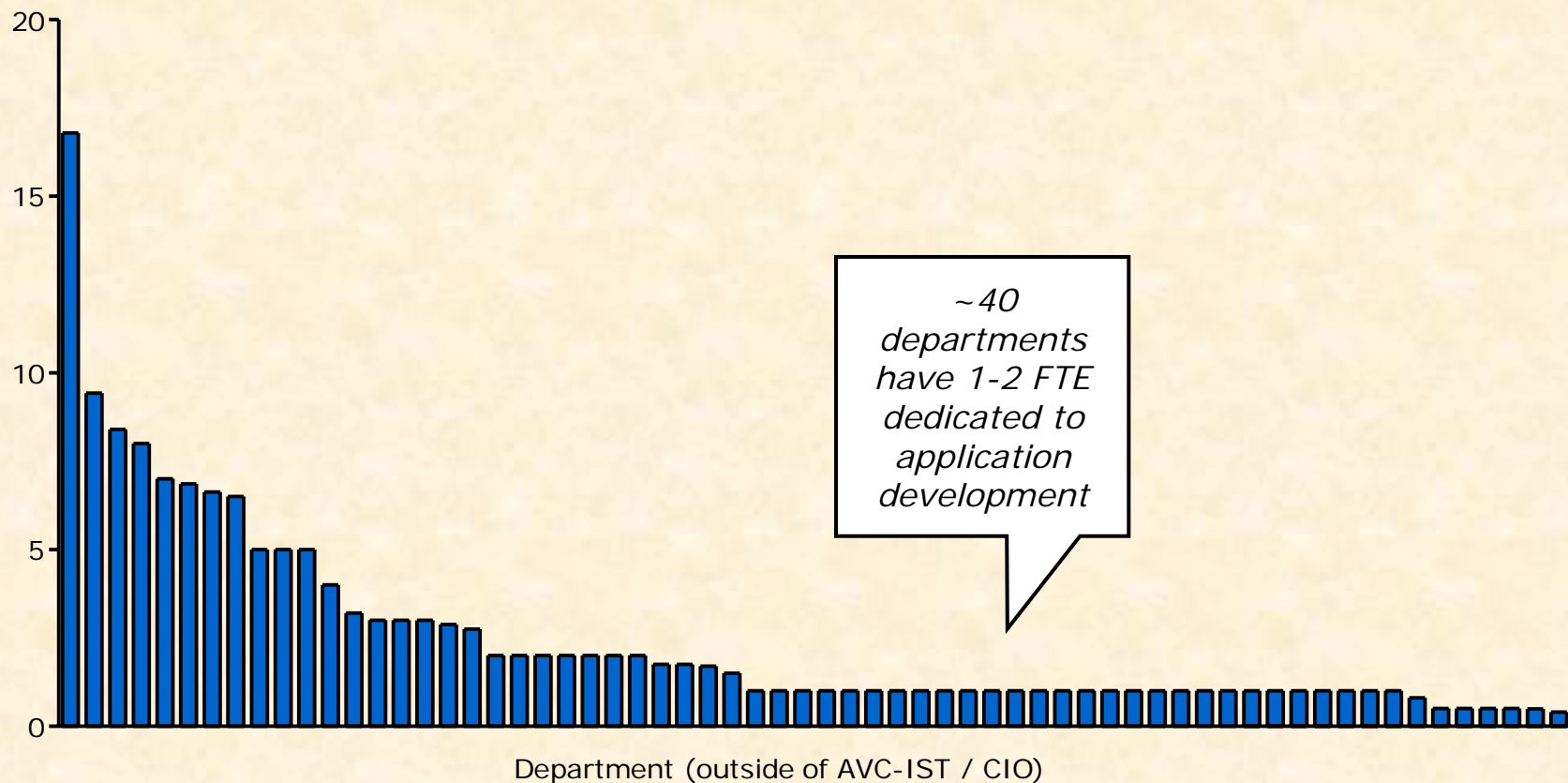
Total = 747 FTEs

IT FTEs by function and control unit



High degree of decentralization across IT functions: App Development example

Number of Application Development Personnel (FTEs) by department (as of 10/7/2009)



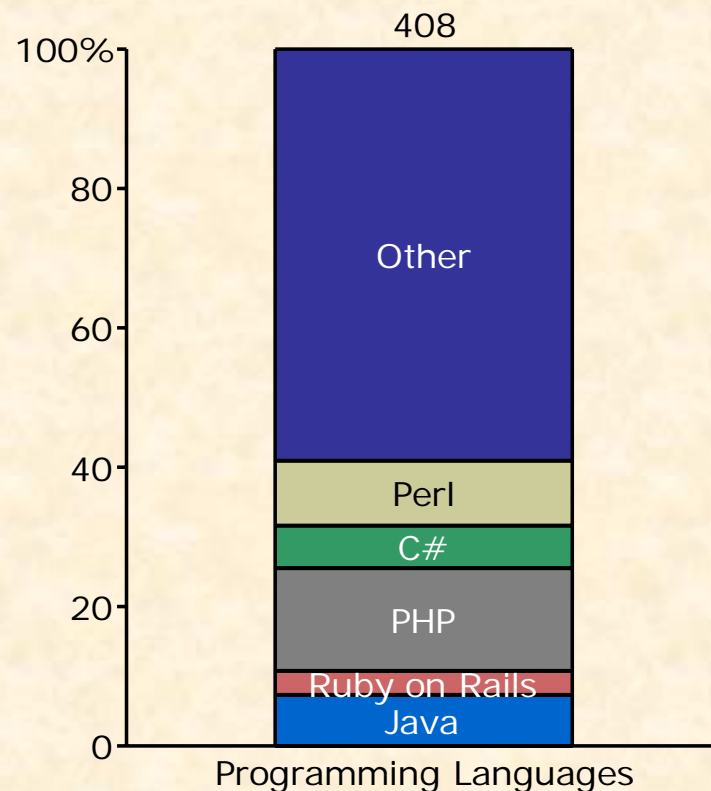
Note: Application Development Personnel include: Application Programmers, Application Programming Mgrs, Application Programming Supervisors; Only departments outside of AVC-IST / CIO with IT personnel as categorized in Career Compass included, out of ~300 depts total
Source: HR Database (Bain-Dataset 20091007V2.xls, data as of 10/7/09)

Internally developed applications have been built in over 20 languages

No prevailing university-wide programming language

Wide range of "Other" languages reported

Applications reported



- WordPress
- Witango
- Visual Basic
- Python
- Paradox
- MS Access
- Matlab
- Lasso
- IBM Universe
- Haskell
- Foxpro
- Flash
- Drupal
- Cold Fusion
- Cobol
- C, C++
- ASP
- 4D



Core problem is then:

Is there a substantially better way to develop, operate, and sustain research museum technologies in higher education?



Better for Whom?

Scholars and curators.

IT, Museums, Libraries.

Campus.

External institutions and Public.



Reminder:

Museum research collections are one instance of more general problem of development and support of e-research or cyberinfrastructure



Enterprise-class expectations...

- Functional expectations from enterprise-class services in banking, search, reservations, etc.
 - Secure, scalable, efficient
 - Aggregate lots of information and behavior
- Institutional demand for access to and functionality around collections and archives information.
 - Aggregation, analysis, or simply *discovery*.
 - Must be simple, scalable, and secure.
- Many experiments with mash-ups:
 - Map mash-ups to visualize the geographic distribution of a dataset
 - Semantic mash-ups to analyze, extract key concepts, or categorize collections w.r.t. a conceptual ontology.

... need Enterprise tools ...

- Traditional developers of technology for these domains are subject-experts, but IT-amateurs.
- Many see the need, but lack the skills and resources to build such a solution
 - Php/perl/MySQL expertise is not up to the task of building a scalable web-services infrastructure.
 - Part time IT and grad students not enough
- Funding model must recognize and support central solution
 - Departmental and research unit funding supports local solutions, rather than a shared, reusable framework.
 - Funding agencies often ready to address this need



... and an Enterprise focus

Functional analysis teams traditionally miss important needs and constraints

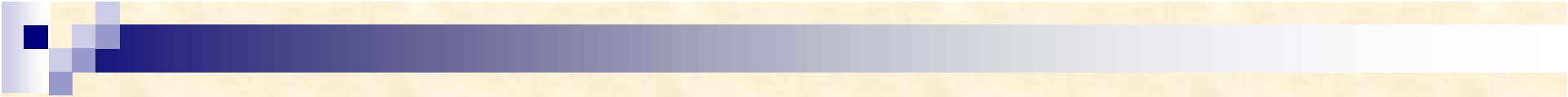
- Focus on the users of an application, to understand what they want it to do (*at least, we hope so*)
- Forget to ask the non-users.
- Ignore the folks who must deploy and support the app

Result is a proliferation of idiosyncratic tools that are brittle, expensive to support, and cannot scale or expand to meet new needs.



The Snowflake Fallacy

- “But we (fill in discipline / department) are unique and thus we must do it ourselves”
- Or, you are too slow and unresponsive, thus we must do it on our own
- There is uniqueness, but a great deal of commonality at multiple levels.



CollectionSpace: The Opportunity

CollectionSpace

CollectionSpace is an open-source, web-based software application for the description, management, and dissemination of museum collections information – from artifacts and archival materials to exhibitions and storage.



 collectionspace

Project Partners

- Museum of the Moving Image, New York
- University of California, Berkeley, Information Services and Technology Division
- University of Cambridge, Centre for Applied Research in Educational Technologies
- OCAD University, Adaptive Technology Resource Centre, Fluid Project

MUSEUM
OF THE MOVING
IMAGE



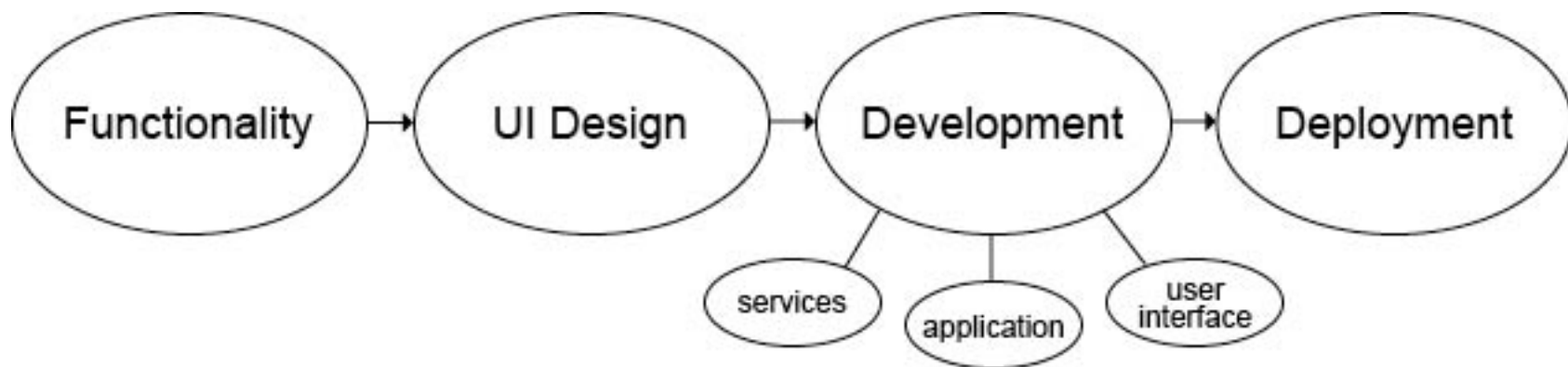
caret.

Fluid

Project Team

The CollectionSpace project team is composed of domain experts, designers, architects, and developers from each partner organization.

Development teams work in cycles to issue regular software releases.



Funding

- The Andrew W. Mellon Foundation, Program in Scholarly Communications and Information Technology
- Institute of Museum and Library Services (IMLS)
- Collaborations with Mellon-funded projects
 - ArchivesSpace
 - OLE Project
 - ConservationSpace
 - Project Bamboo
- Considerable local (UCB) investment!
- Funding is for *development*, not operations, or sustainability



Community Source

CollectionSpace is based on the community source model:

“A hybrid model that blends elements of directed development, in the classic sense of an organization employing staff and resources to work on a project, and the openness of traditional open-source projects like Apache...the distinguishing feature of the Community Source Model is that many of the investments of developers’ time, design, and project governances come from institutional contributions... rather than from individuals. The project often establishes a software framework and baseline functionality, and then the community develops additional features as needed over time.”



Community Source

- Benefits of Open Source +
- Structured and coordinated development process
- Designed WITH user community
- Reduced total cost of operations
- Doesn't scare our colleagues
- *But:* Incurs overhead for coordination, communications



Project Timeline

2007: Initial planning, partner meetings

2008: Community design workshops, high-level architecture, list of candidate services

2009: Initial wireframes, tech integration, first set of core (end-user) procedures

2010: 1.0 version ships

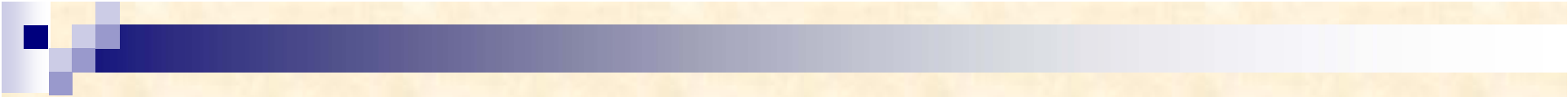
2011: 2.0 version ships, early adopters

2012: SaaS support, sustainability model



CollectionSpace Pilot Deployments

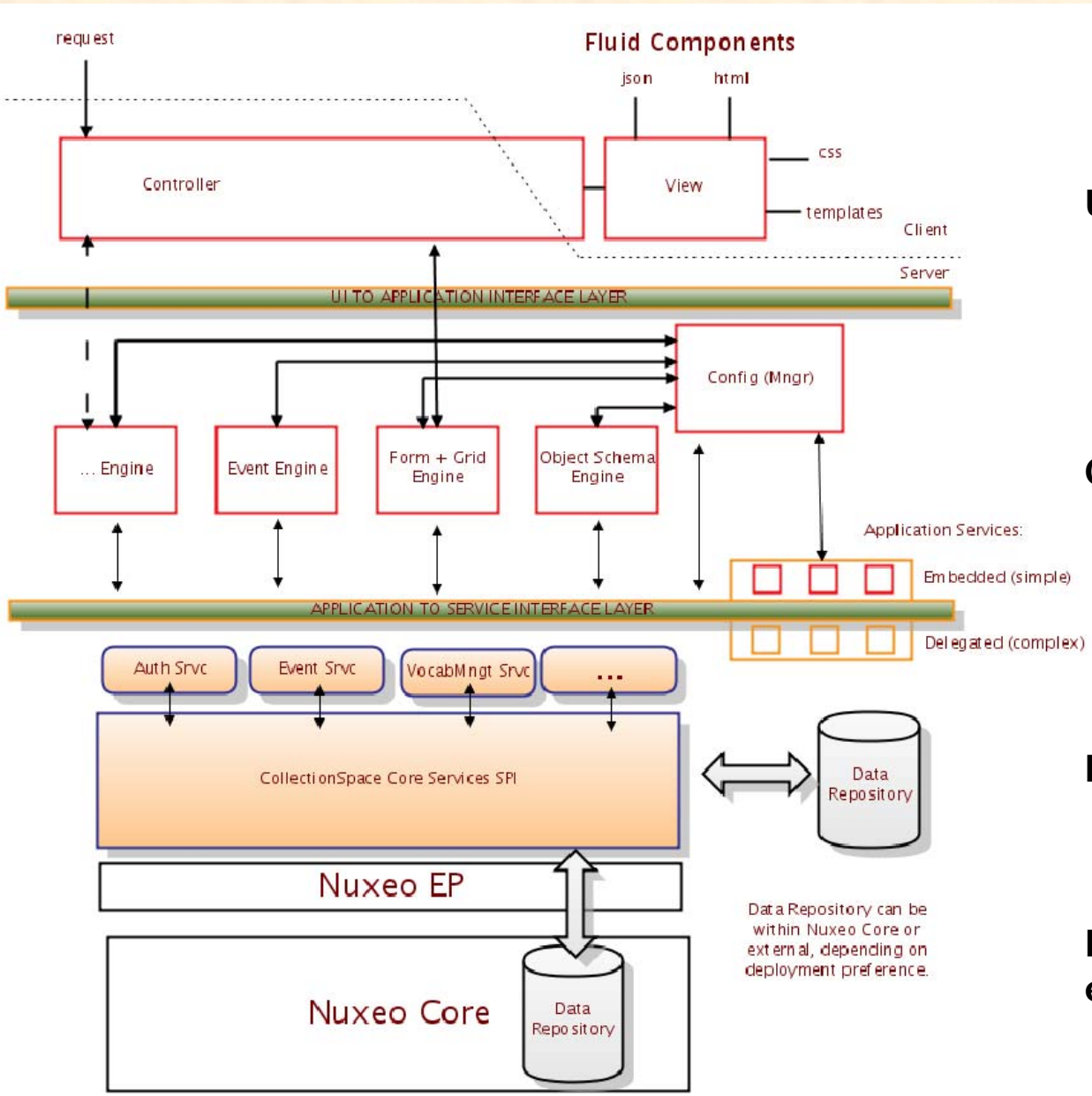
- Working on pilot deployments to gain experience
 - Domains from Anthropology to Life Science to Cultural Heritage
 - Stand-alone as well as hosted deployments
- Developing best practices with tools for metadata migration
- Building templates for initial domains
 - Adaptations, extensions contributed to CollectionSpace community
 - Contributions from community ease future deployments
 - Community provides forum for discussion/sharing experience
- Expanding deployments across a range of domains
- Developing pilots of SaaS hosting model



Architecture and Technology

A Web-Oriented Architecture

- No exotic technologies: just the Web
- HTML, CSS, and JavaScript
- Familiar and extensible
- Clean, simple URLs + useful data feeds (RESTful APIs)
- Built using
 - Fluid's Infusion application framework (jQuery-based)
 - RESTful APIs exposing XML and JSON
- Flexible and Accessible
 - Can accommodate diverse user needs
 - Works well with the keyboard, other assistive technologies
 - Accessible, but still rich and dynamic!



U. Toronto/Fluid

Cambridge/CARET

Berkeley/IST-DS

Nuxeo (Apache, etc.)



Skills inventory for C-Space

- Core framework coding
 - Java, Tomcat, Spring, Nuxeo, SOA/ROA/REST
 - PM/Dev tools: Ant+Maven, Wiki/Jira, SVN/Git, etc.
- Service definition and development
 - Domain expertise, experience
 - XML/XSD, Java, eventing/messaging, workflow, etc.
- Deployment, Customization and Extension
 - XML/XSD, JSON, for schema and for app. configuration
 - Javascript, jQuery, Fluid/Infusion for app code
 - HTML/CSS and Infusion templates for presentation

CollectionSpace UX Goals

- A holistic product
- Designed by museums, not technologists
- Easy to use, but not simplistic
- Accommodates your workflow & collection
- Accessible to a wide variety of user needs

How We're Making it Easy

The screenshot shows the 'collectionspace' interface for editing a record titled 'GIRL IN A BOX'. The record ID is '2004.002.001'. The interface is divided into several sections: 'Object Identification Information' (collapsible), 'Object Description Information' (collapsible), 'Media Snapshot', 'Integrated Authorities & Vocabularies', 'Related Procedures', 'Related Objects', and 'Related Collections'. The 'Object Identification Information' section includes fields for 'Titles', 'Object name', 'Identification Number', 'Other Numbers', 'Components and Range', and 'Responsible Department'. The 'Object Description Information' section includes 'Copy number' and 'Edition number'. The 'Media Snapshot' section shows a timestamp 'IMG9201.1F'. The 'Integrated Authorities & Vocabularies' section shows a list of authorized terms. The 'Related Objects' section shows a list of related objects. The 'Related Procedures' and 'Related Collections' sections are currently empty. The bottom toolbar includes a search box, a 'Next' button, a 'now editing:' indicator showing '1 / 5 new records', a 'select a shortcut:' dropdown, and 'Revert', 'Cancel', and 'Save' buttons.

1. Each information group can be collapsed to decrease screen clutter

2. Simple radio buttons allow users to choose which value of a repeatable field should be considered "primary"

3. Markers in each field denote behavior - whether the field leads to a predictive text or dropdown pulled from a controlled list or authority file

4. Repeatable fields can be added with a press of a button

5. Data entry screens each include a toolbar at the bottom that simplifies searching and saving

6. Links to related procedures, objects, and collections can be created and managed

7. An integrated authorities list gives an index to all the authorized terms referenced in this record

8. The time stamp for the last save or auto-save is displayed. At any time, changes can be reverted or cancelled

Out of the Box Experience

collectionspace

My CollectionSpace Create New Find and Edit Report Administration

2004.002.001 GIRL IN A BOX

VIEW LOG NOTE PREVIEW

Cataloging Condition Conservation Location & Movement Valuation & Insurance Rights Media

Object Identification Information

Titles

Title	Title language	Title translation	Title type
Girl in a box	English	n/a	

+ Add a row

Object name

Object name	Currency	Level	System	Type	Language	Note

+ Add a row

Identification Number

2009.002.001 [Select number pattern](#)

Other Numbers

Number	Number type

+ Add a number

Components and Range

Total Number of items: 16 [Generate sub numbers](#)

Sub number		
1	2004.002.001a	Create record
2	2004.002.001b Queen Chess Piece	Create record
3	2004.002.001c	Create record

Responsible Department

- select a department -

Brief Description

Distinguishing Features

Comments

Object Description Information

Copy number: Edition number:

Find within this record: Next

now editing: < 1 / 5 new records >

select a shortcut: - Select -

Revert Cancel Save

Media Snapshot

PREV IMG9201.M NEXT

Integrated Authorities & Vocabularies (7)

Authority	URI	URI
2004.002.001	2004.002.001	2004.002.001
2004.002.001	2004.002.001	2004.002.001
2004.002.001	2004.002.001	2004.002.001
2004.002.001	2004.002.001	2004.002.001
2004.002.001	2004.002.001	2004.002.001
2004.002.001	2004.002.001	2004.002.001


Related Procedures (0) + add

Related Objects (5) + add

Object	URI	URI
2004.002.001	2004.002.001	2004.002.001
2004.002.001	2004.002.001	2004.002.001
2004.002.001	2004.002.001	2004.002.001
2004.002.001	2004.002.001	2004.002.001
2004.002.001	2004.002.001	2004.002.001

Related Collections (0) + add

Customized Museum Experience



Dashboard

[Administration](#) | [Preferences](#)

[> Advanced Search](#)

Create New
Find and Edit
Report
My CollectionSpace

VIEW LOG NOTE PREVIEW
< PREVIOUS 1/5 NEXT >

NEW OBJ. 2004.002.001 Draft saved at 3:06pm

Cataloging
Condition
Docs/Media
Location

Object Identification Information

Object Title

Accession Number +

Artifact Class

Work Type

Extant

Collecting Category

Brief Description

Upright Moviola Model D 35mm table top editing machine. Equipped to handle one reel of motion picture film (without sound), this early Moviola was probably powered by a sewing machine motor. It uses intermittent sprockets to run film through the machine and is shown with an electrical power cord, two wire

Historical Notes

Founded by Dutch-born Ivan Serrurier, the Moviola Manufacturing Company sold its first table top Moviola Editing Machine to the Douglas Fairbanks Studio in 1924. Moviolas made it possible to synchronize sound and picture, or edit them independently, which helped the Moviola become standard equipment for sound

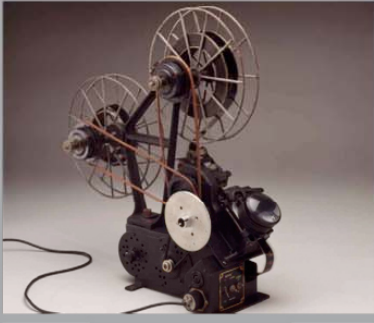
Historical Note Sources

Stuart Klawans, Morris Engel and Ruth Orkin, Pioneer Independent Filmmakers, to Attend Special Cineprobe Evening November 20, New York, NY: Museum of Modern Art, November, 1984.

Overall Dimensions

Date	Dimension	Unit	Description	Note
	Height	in		
	Weight	lb		
	Depth	in		
	Narrative	n/a		

Media Snapshot



◀PREV Media Name: Photo 1 NEXT▶

▼ **Related Organizational Records (0)**

▼ **Related Procedures (1)**

ID	Type	Created on
2004.002	Intake	04.04.09

▼ **Related Objects (0)**

No related objects yet



Configuration and Customization

- Configuration of existing services, schemas
 - Which services are of interest for this deployment
 - Which fields in schemas to use, how to label them
 - Validation rules, patterns, for field values
 - Roles, access policies, for pages, fields, etc.
 - Vocabularies, name authorities, etc.
- Customization of schemas, application
 - Pageflow, graphics, look and feel of application
 - Local schema extensions
 - Application extensions to integrate other services, etc.

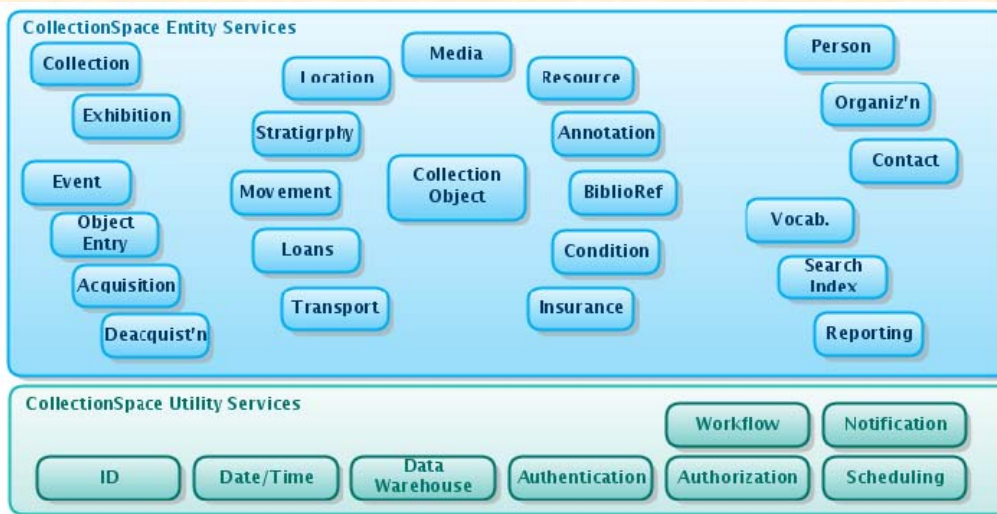


Leveraging ECM

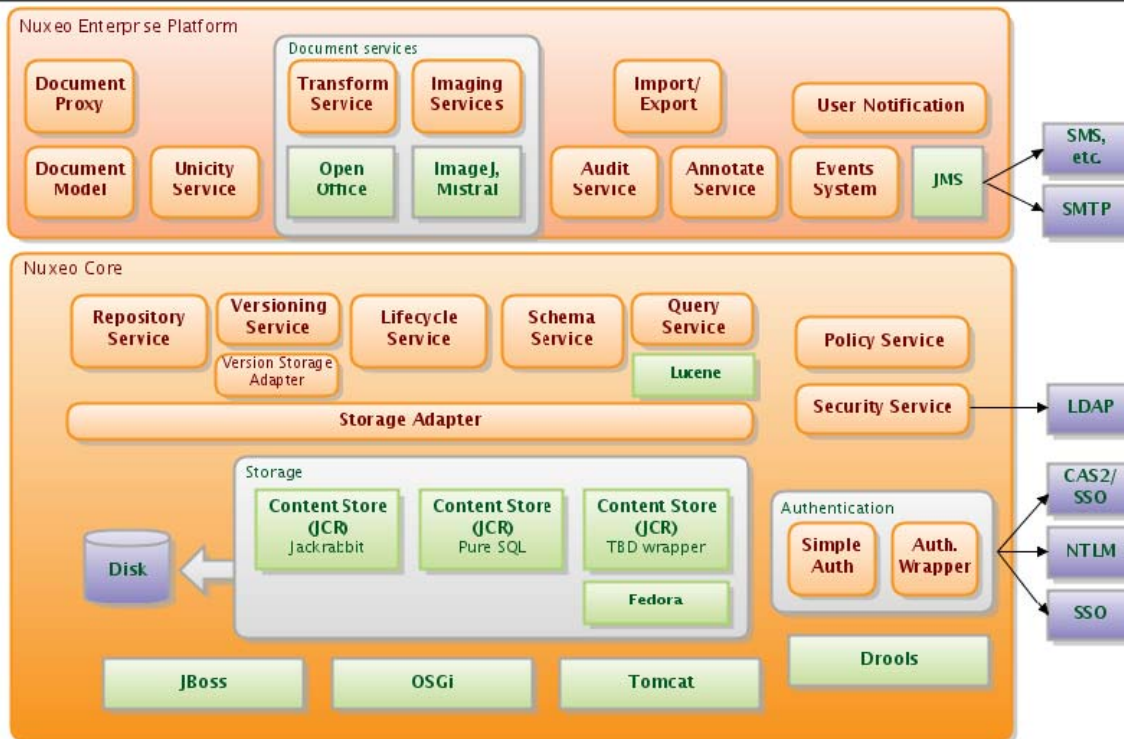
- Prevalence of content-centric applications
- Enterprise Content Management (ECM) is a natural platform upon which to build
 - Re-use is a necessity
 - Provides rich, flexible functionality
- ECM \neq WCM (web-content management)
 - Drupal has its uses; this ain't one of them.
- CMIS (OASIS) emerging as abstraction layer

Services stack

C-Space
Services



Nuxeo
Platform
Services





Services Platform as Strategy

- Web-services approach enables mashups
 - Also, new applications now yet envisioned.
- Services approach allows re-use across multiple domain-specific applications
 - Many collections do cataloging, accession, loans, controlled-vocabulary, etc.
 - Each domain has specific needs, but share much
- Services approach allows for different compositions for different domains
 - Art History may not need Stratigraphic-location