CHIME: A Metadata-Based Distributed Software Development Environment

Gail Kaiser
Programming Systems Lab
Columbia University
http://www.psl.cs.columbia.edu

Introduction

- Even small software development efforts may involve hundreds of artifacts and tens of developers
  - Sheer volume makes it difficult to find what one is looking for, or what one should be looking for
  - Hard for new project members to get started and come up to speed
- Other research areas (beyond traditional software engineering) may offer some hope
  - Open Hypermedia Systems
  - E-Business and other Web technologies
  - 3D Information Visualization
  - Multi-User Domains (MUDs)

CHIME

Columbia Hypermedia IMmersion Environment

- Users “walk around” a 3D virtual world semi-automatically generated from project metadata
  - New team members collaborate with “old-timers” who may be in different physical locations
  - Users are aware, not just of other users’ presence, but of what they’re doing with which artifacts
- XML-based framework enables artifacts to reside in their original locations and be extracted on demand
  - Source Code and Configurations in SCM system
  - Design Materials in UML tool’s repository
  - Email archives on Corporate Intranet


CHIME 2 : sdossick
CHIME
Inventory
Team
Users
NRC
History
NUCLEAR
REACTORS
NUCLEAR
REGULATORY
COMMISSION
CHIME 2 chat
Current room: Start
Room users: sdossick
Current room: NRC
Room users: sdossick

Conceptual Model

- ***Groupspace***
  - Persistent shared virtual information space
  - Represents artifacts and tools that create, modify and maintain them
  - May introduce metadata - hyperlinks, attributes, annotations - not resident and unknown in native content or repository

Conceptual Model (2)

- ***Groupview***
  - Scalable team-oriented user interface paradigm
  - Zooms out from WYSIWIS and shared desktop
  - Pans away from talking heads video conferencing
  - Zooms in to selective presence and awareness

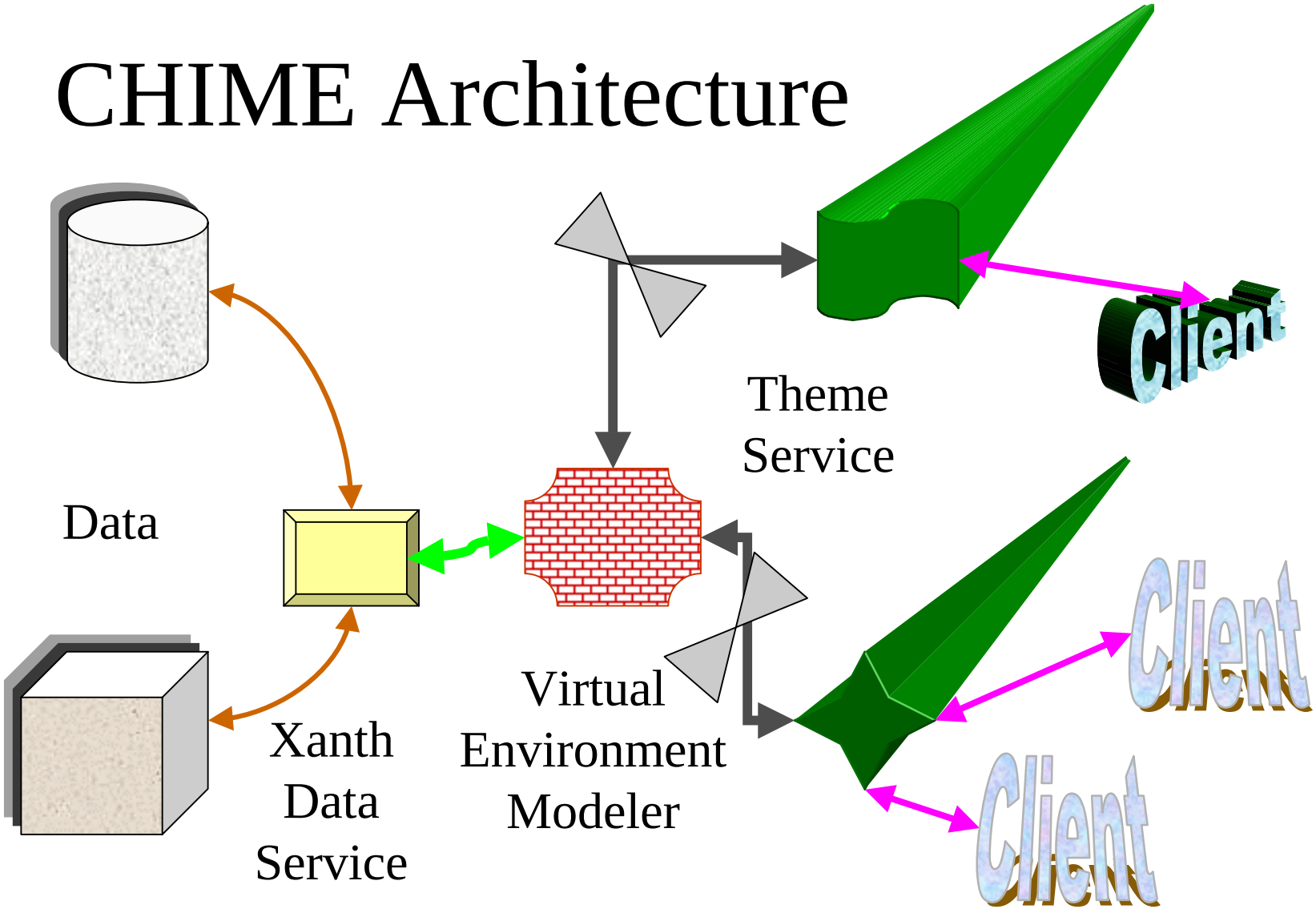
Conceptual Model (3)

- ***Software Immersion***
  - 2D or 3D graphical project-specific “portal”
  - Co-located users “see” each others’ avatars in context
  - Virtual world behaviors map conventional and groupware tools to avatar actions and proximity
  - Virtual world spatial layout and contents defined by relationships among artifacts

Examples

- What we can do now: Code module becomes a room, source files become furnishings or alcoves in the room – depending on desired granularity. Corridor might link a module's room to rooms containing related design docs, test plans, etc. - geometry not necessarily Euclidean.
- What we'd like to do 1: Requirements workbench becomes a room, linked by alarmed doorway to design workbench room. Wall transparency vs. opaqueness determined by communication levels.
- What we'd like to do 2: Run-time architectural components become rooms, linked according to run-time configuration and communication topology. Inter-component messaging shown as scurrying bots.

CHIME Architecture



CHIME Architecture (2)

- Xanth Data Service
  - Data organized into multi-rooted tree hierarchy of XML dataElements
    - Each dataElement describes a piece of data residing in a remote repository
  - Xanth maintains an XML document which completely describes the contents of the Groupspace

CHIME Architecture (3)

<dataElement
name=“README”
id=“1000”
protocol=“http”
server=“library.psl.cs.columbia.edu”
port=“80”
path=“/linux-2.0.36/README”
hidden=“false”
parent=“0”
behavior=“GET” />

CHIME Architecture (4)

- Xanth Data Access Modules
  - Implement retrieval protocols needed to communicate with data repositories
  - May provide “behaviors” to perform actions on the data
    - E.g. for HTTP, behaviors are GET, POST, PUT
- Xanth Link Service
  - Imposes typed, n-ary, bi-directional hypertext links among dataElements

CHIME Architecture (5)

- Virtual Environment Modeler (VEM)
  - Parameterize each dataElement according to a set of Virtual Environment Types
  - Base Types: Component, Container, Connector
  - VEM Types extensible for customization to underlying data
  - VEM does not “hardwire” any user interface paradigm

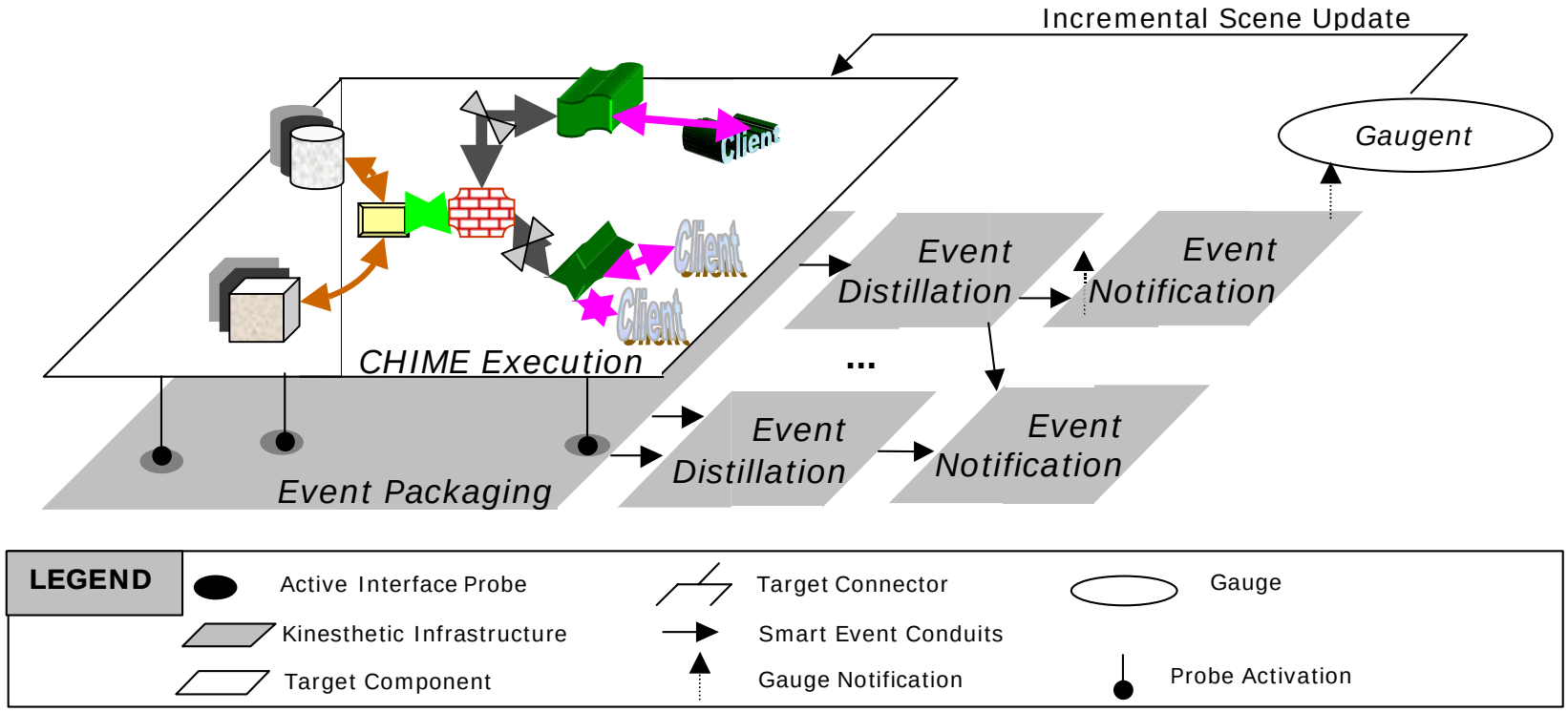
CHIME Architecture (6)

- Theme Service
  - Maps VEM (sub)types to specific scene graph elements from theme plugins
  - Theme Plugins
    - Downloaded from server at runtime for client rendering
    - Not necessarily 3D, could include icons, text, etc.
  - MUD Service
    - Tracks user locations, handles user authentication, communications between users (e.g., chat), etc.

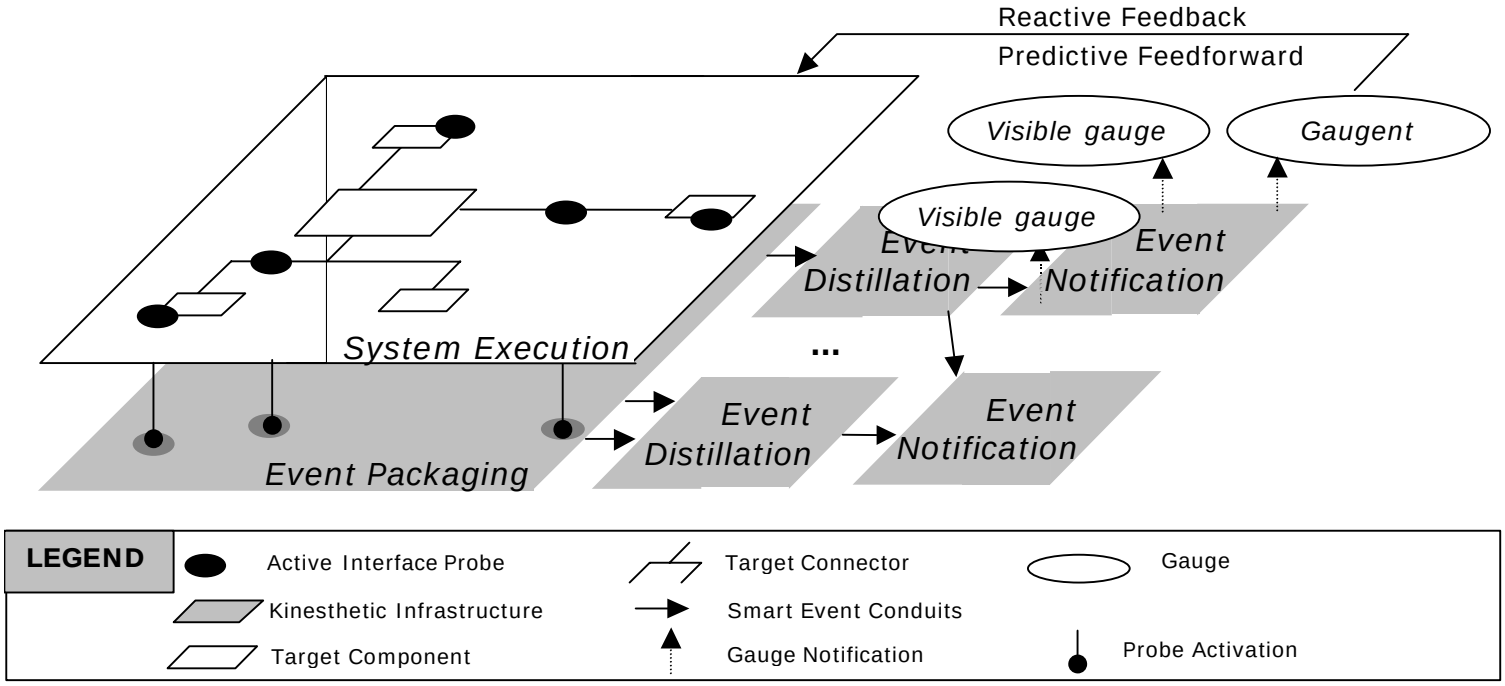
Immediate Challenges

- Remote repositories for Groupspace data
  - May not natively provide necessary metadata or transfer protocols
  - May not support locking, transactions, undo/redo
  - May be temporarily unavailable
- Artifacts added, modified, deleted, merged, split, moved in underlying Groupspace
  - How to reflect incremental changes in Groupview while retaining spatial metaphor
- Complexity of underlying software project
  - How to design a Software Immersion “theme” that provides a *useful* spatial metaphor intuition

KX Monitoring Architecture v. 0



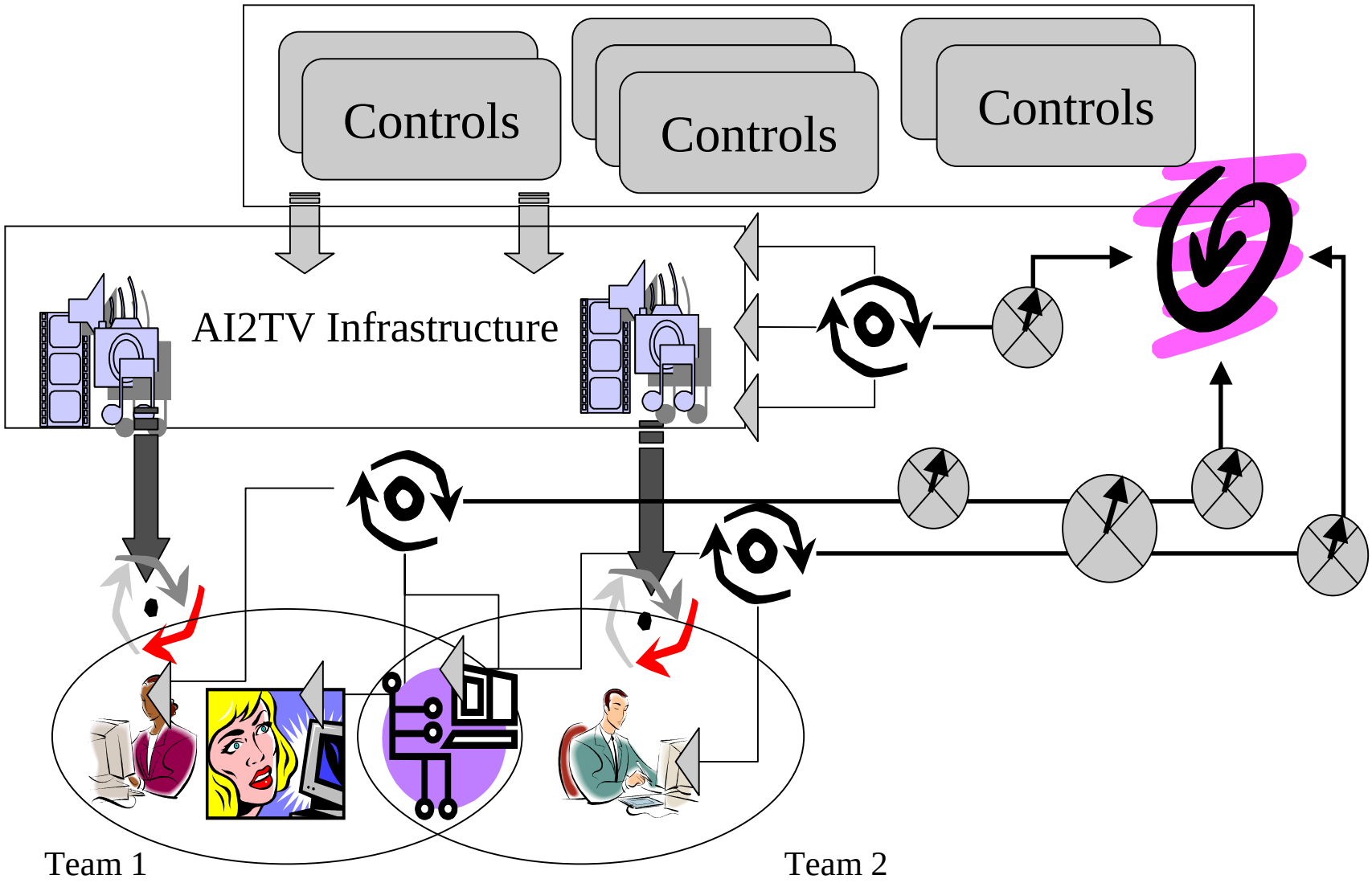
KX Monitoring Architecture v. 1



Kinesthetics eXtreme
“Continual Validation” infrastructure

- “Smart” Events encoded in **FleXML**
  - Stream-oriented variant of XML
  - Discoverable and Composable DTDs (or XML schemas)
- XML-based Universal Event Service
  - Events propagated among allied publish/subscribe “active connectors”
  - Partially ordered event stream handling by programmable FleXML meta-processors
  - Dynamic event recognition criteria and processors deployed via mobile agents from push caches

Another KX Application



Contributions (of CHIME)

- Framework for developing project-specific team-oriented software engineering portals (proven portal building by “outside” students)
- Intuitive 3D virtual world GUI exploits popularity of social MUDs and first-person “shooter” gaming
- Scalable wrt kinds of artifacts (prior Xanth work), number of artifacts (proven, e.g., Linux), users and dispersion (speculated)
- Should support process/workflow and business information immersion as well as Software Immersion for e-business portals
- And now here are the demos…

enable (vt) : *to make possible, practical, or easy*

PROGRAMMING SYSTEMS LAB
COLUMBIA UNIVERSITY

http://www.psl.cs.columbia.edu/