Why Would You Do This?

Developing a Community Source Student Services System

Richard Spencer
University of BC

JA-SIG 2006 Winter Conference, Atlanta, GA

December 4, 2006

Goals for enterprise systems

- Scalable, rule based, self-service processes
- Strong focus on needs of end users (not just staff in central departments)
- Departmental responsibility for services
  - HR, Finance, Plant, Admissions, Registrars, etc. must make sure systems deliver high levels of service and efficiency
- An architecture that:
  - supports complete academic and business processes
  - allows business processes to easily span systems
  - makes it easy to modify business processes

Use IT to help end users achieve their goals

A brief history of student systems

- BC
  - paper based processes, forms, information silos
  - the customer had to help us run the institution
- SRS
  - batch systems, on-line records, flat files, reports
  - developed in-house
- SIS
  - support for core processes
  - more work for other staff, some support for customers
  - ERPs
- ***SSS***
  - *Next generation enterprise system*
  - *Community source development*

Origins of the SSS

- Meeting at Educause in Orlando, October, 2005
  - 35 people from 19 institutions, 3 vendors, & other organizations
- Mellon Foundation funded planning study
  - Goals of study
    - what is the level of interest in an open source SSS?
    - what is the need for an open source SSS?
    - are there existing applications to use as a base?
  - Participants
    - University of Indiana, Georgetown University, San Joaquin Delta College, UBC, consultants and others
  - Consultation
    - meetings at JA-SIG and Sakai conferences
    - workshops, focus groups, consultation with vendors

Need for a next generation system

- add functionality to existing systems
  - ERPs can't do everything
  - re-use some existing functionality

Delaware:

- housing
- dining
- course approval
- judicial referral
- course & faculty evaluation
- advising notes

Indiana

- course trading

The Student Services System vision

A new system that is:

1. person-centric
2. learning, learner and institution agnostic
3. modular, technology neutral – runs everywhere
4. easy to support new business processes
5. an open source, community source, project

use new technology to realize a compelling new vision

1: Person-centric support

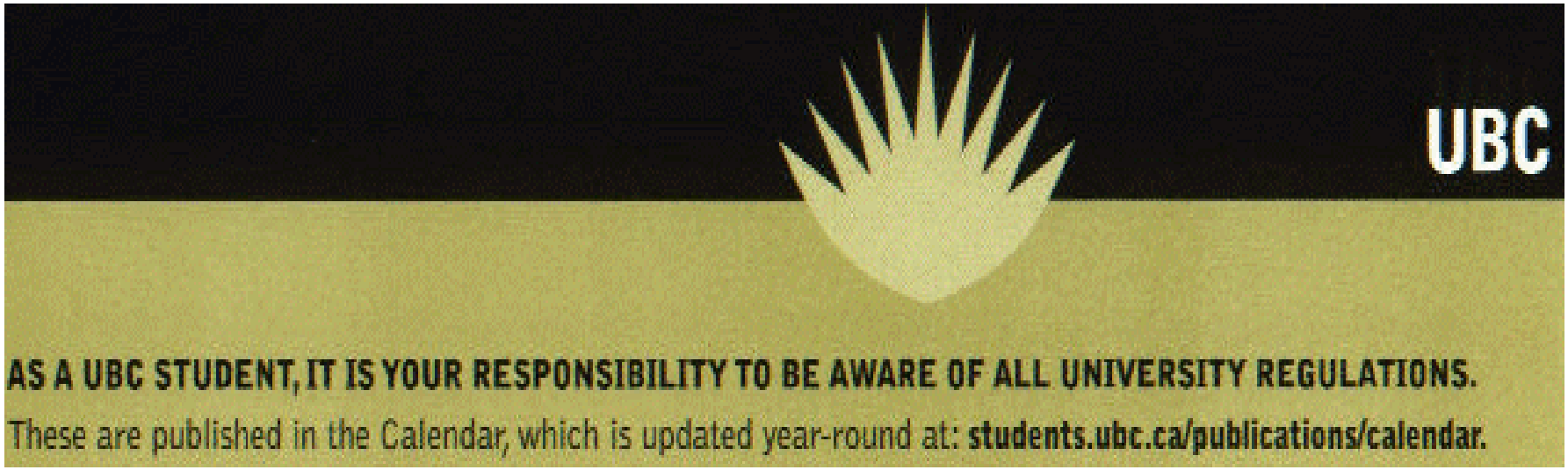
- help users develop learning plans and achieve goals
- use what we know when providing services
- suggest valid choices, apply and explain rules
- anticipate people's needs
- integrate processes
- make tasks simpler and easier
- online solutions are reliable and trustworthy

a "concierge" to support students, faculty and staff

1: The concierge

- Anticipate people's needs
  - review accomplishments and plans
  - suggest desirable or required actions
  - present the applications required to complete them
- Apply and explain the rules
  - know all the relevant institutional rules and requirements
  - review user's situation, monitor actions being taken
  - present choices that satisfy rules and requirements
- Integrate processes
  - present more than one application as required
- Learn from experience
  - use artificial intelligence to make the concierge smarter

“it is your responsibility....”



UBC Calendar insert, 1999

1: Person-centric identity systems

- A person is a high level entity
  - a person has roles, group memberships, etc
    - student, employee, faculty, alumni are roles
- Any person can establish an on-line identity
  - easy an to get an account with an authenticator
  - verify IDs as people build their relationship
  - federation of existing IDs should be encouraged
- Separate authentication and authorization
  - authentication confirms the owner of the ID has logged in
  - authorization is based on person's relationships and roles
    - access to resources may initially be very restricted

an on-line ID is for access, as well as security

2: Learning and institution agnostic

- support all types of learning
  - including non-credit and non traditional
- the learning unit entity
  - course; single lecture in a course; 15 minute student presentation in a course
  - participation in community service
  - any activity that the student wants to include on a formal or co-curricular transcript
  - a “learning unit number” is like a SKU...
- learning results, learning plans
  - entities that represent transcripts and programs

don’t restrict what people and institutions can do

2: Flexibility

- flexible time frames
  - measure time in years, days, hours, minute and second
- support all institutions and types of program
  - 2 year colleges to doctoral/research
  - non-credit, non-traditional
  - no built in assumptions about programs, program approval, etc.
- international
  - easily handle different languages and conventions

minimize or eliminate system constraints

3: Modular, technology neutral

- different institutions can build applications that will work together
- applications can use different technologies
- applications can be integrated with existing systems
- open source and commercial applications can be combined
- a critical mass of applications will deliver a complete next generation system

deploy what you need, when you need it

3: Preliminary plan for applications

- customer contact
- admission
- curriculum development
- enrolment
- degree audit and academic evaluation
- awards
- student financials
- scheduling

4: Easy to change business processes

- portal for user interface
  - standards based
  - flexible and powerful
- rules engines for internal process logic
- workflow for end-to-end business processes
  - processes can cross systems
- encourage and support innovation and change
- it's OK to customize.....

your practices, not someone else's "best practices"

4: Portal wish list

If all services are delivered through a portal:

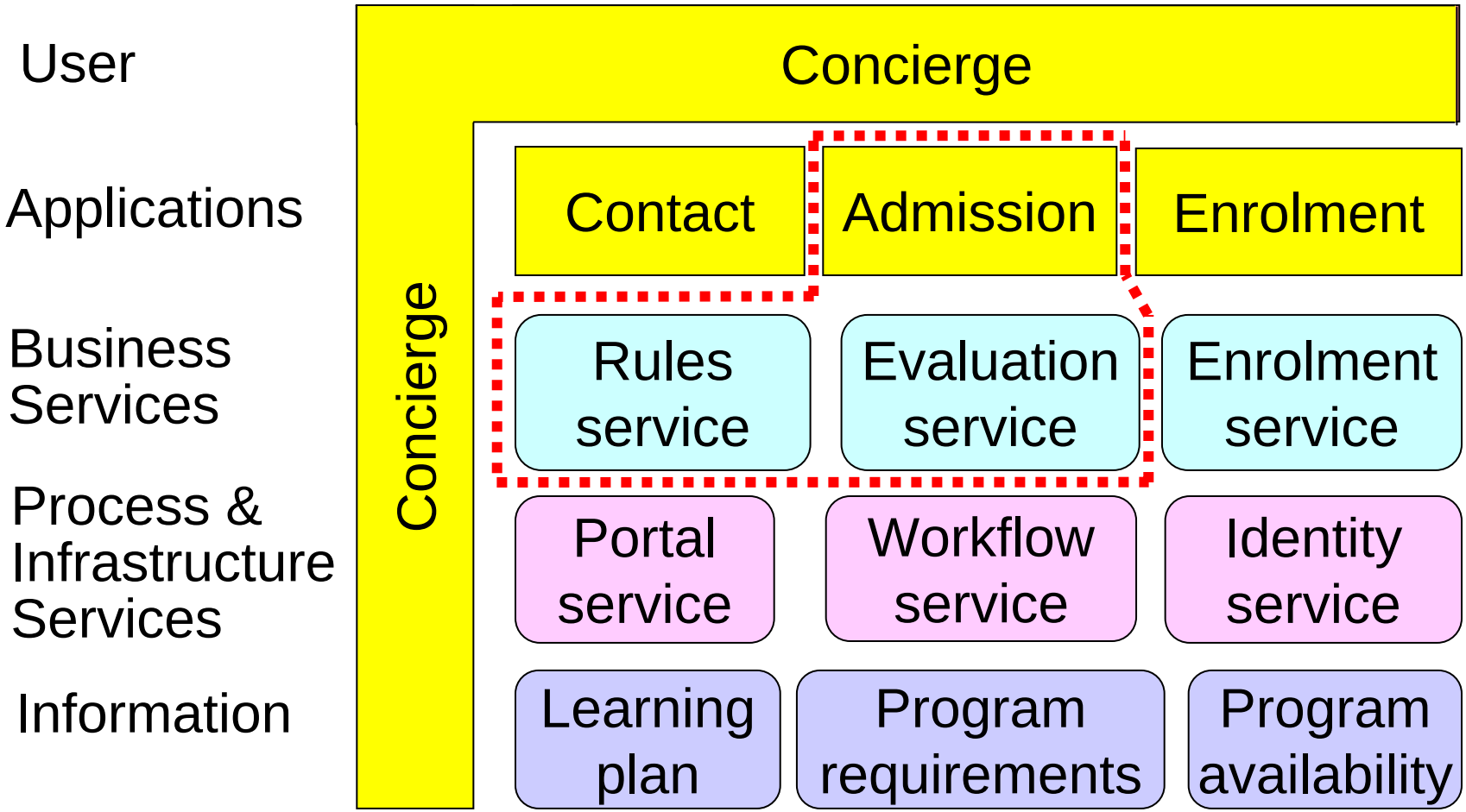
- support for single signon, with multiple personas
- tabs and channels presented can be role based
- the system can control some tabs
- system can provision tabs and channels
- messages can be delivered through the portal
- personalization – system can configure the portal to reflect situations and events as they affect a user
- customization – user can create an interface that works for them

4: Services and SOA

- business processes are decomposed into services
- services:
  - are autonomous, agnostic, and reusable
  - are defined by service contracts and interface definitions
  - use standard data models and XML schemas
  - include:
    - process or control services (workflow, orchestration)
    - business services (rules engines)
    - infrastructure services (identity, portal, database services)
- Service oriented architecture:
  - uses web services to loosely couple components
  - supports business processes that cross applications

Services and SOA makes process change easier

4: Conceptual system architecture

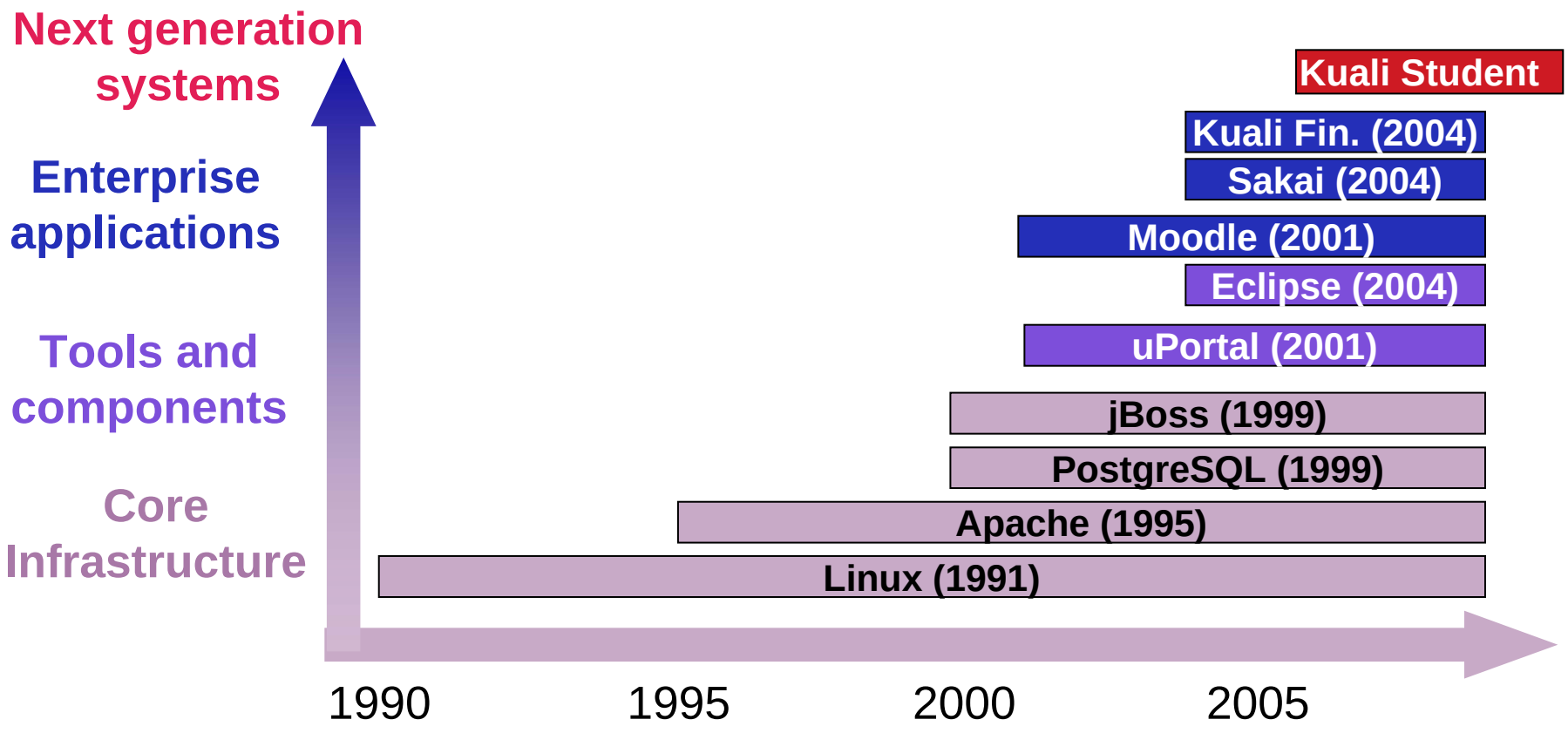


5: Community source development

- investors:
  - establish a board, functional and technical councils
  - agree to develop and implement specific applications
  - have direct input into functions and features
  - share resources
  - use common standards, schemas and interface definitions
  - work together on service oriented analysis
  - build and share services
  - meet agreed development and delivery schedules
- commercial installation and support is encouraged
- larger community ensures sustained development

build on uPortal, Sakai and Kuali experience

5: Open and community source experience



Open vs. community source

Open source

- Open membership, voluntary participation
- Large developer community
- "Voluntary hierarchy"[1] decides priorities & projects
- Corporate contributions OK
- Local development can lead to forking and different versions

Community source

- Membership in an entity, with defined roles and funding
- Smaller development community
- Formal project management methodology and project charter
- Institutional contributions required
- Locally developed components should be compatible

- Source code can be freely reviewed, modified, and redistributed

---

1. Felix Stalder "Open Source Projects as Voluntary Hierarchies", review of "The Success of Open Source", by Steven Weber

The benefits of community source

- Sharing
  - more resources on each project, higher productivity
- Functionality
  - investors have direct input into functions and features
- Innovation
  - investors drive innovation
  - commercial partners can also innovate
- Future path
  - community can ensure sustained development
  - open source model begins to work
- Support
  - commercial installation and support is encouraged

Current Kuali Student Founders

- University of British Columbia
- Carnegie Mellon University
- University of Maryland
- San Joaquin Delta College
- UC Berkeley

Next steps

- Finalize project governance, investors
- Complete project plan for a complete student system
- Identify core Kuali Student applications
  - service oriented analysis of key business processes
- Finalize a reference technical architecture
- Continue to develop the vision
  - model the “concierge” concept
- Develop the entity and data models
- Submit a funding proposal to the Mellon Foundation
- Begin developing applications

Challenges

- Entity models, standards and schemas
- True service analysis and orientation
- Blending stand alone applications with service orientation
- Web services for loose coupling
- Combining modules developed at different schools
- Combining open source and commercial components
- Using commercial service providers to implement and support systems and system components
- **Flexible, user focused, business processes**

More information

http://student.osnext.org/

http://educationcommons.org/projects