Open Knowledge Initiative

UIS Java Hour
Lance Speelmon & Randall Embry
7/15/2003
What is OKI?

“The Open Knowledge Initiative™ is a collaboration among leading universities and specification and standards organizations to support innovative learning technology in higher education.

The result of this collaboration is an open and extensible architecture that specifies how the components of an educational software environment communicate with each other and with other enterprise systems. O.K.I. provides a modular development platform for building both traditional and innovative applications while leveraging existing and future infrastructure technologies.

O.K.I. is designed for broad adoption in the higher education domain. It provides a stable, scalable base that supports the flexibility needed by higher education and commercial developers of educational software.”
Why Do We Care?

- If applications are written to the OKI specifications, then sharing and reusing applications becomes easier.
- If we can easily share pieces or even entire applications, a significant cost savings can be realized.
OKI Architecture
OKI Timeline

O.K.I. Timeline

Phase One
- Initial Core Service Development
- Reference Implementations
- Applications

Phase Two
- R&D of New Interoperability Specifications
- Maintain and Refine Core Services
- Developer Community Coordination and Support
- Vendor Engagement
- Adopter Engagement

2001 2002 2003 2004 2005 2006
OKI OSID Interfaces

- Assessment
- Authentication
- Authorization
- ClassAdmin
- DBC
- Dictionary
- DigitalRepository
- Filing
- Hierarchy
- Logging
- Scheduling
- Shared
- SQL
- UserMessaging
- Workflow
Authentication OSID

**Functional Description**
The Authentication OSID gathers required credentials from an agent, vouches for their authenticity and introduces the agent to the system.

**Why important**
The Authentication OSID permits an application to abstract the authentication process without having to manage the details of the underlying authentication service.

**Use cases**
An application that permits users to log in remotely must be able to verify that they genuinely represent the user they claim to be.
Authorization OSID

**Functional Description**

The Authorization OSID allows an application to establish and query a user's privileges to view, create, or modify application data, or use application functionality.

**Why important**

Applications that can change Enterprise data need to manage a user's access to that data. An application must provide a fine degree of authorization granularity to reflect the complexity of a user's interaction with an application.

**Use cases**

A graduate student TA has the system privileges to change the grades of the students in the course section that he teaches, but not the privilege to change his own grades.
The Digital Repository Open Service Interface Definition covers storing and retrieving digital content, referred to as Assets, as well as information about the Assets. Assets, examples of which include: documents, course material, assessment item, images, video, audio, etc, reside in Digital Repositories which have names and descriptions and which support a specific set of Asset Types. Digital Repositories are themselves organized by the Digital Repository Manager that keeps track of repositories and supports certain operations such as searching for Assets across repositories. Associated with each Asset Type are InfoStructures that define the format of information comprising the Asset or information describing the Asset. An Asset can have content as well as InfoRecords, which are data in the format defined by the Asset’s InfoStructure. Assets may contain other Assets.
Hierarchy OSID

**Functional Description**
The Hierarchy OSID manages parent-child relationships among elements. In addition to simple tree structures, the OSID supports hierarchy that are recursive and have nodes with multiple parents.

**Why important**
Parent-child relationships are fundamental structures that effectively model a variety of enterprise data.

**Use cases**
User authorizations are usually stored as a hierarchy.
Filing OSID

Functional Description
The Filing OSID provides platform-independent means to handle files arranged in simple hierarchical containers.

Why important
Most applications have occasion to manipulate their data through the use of files in some sort of file system.

Use cases
Homework assignments could be submitted and stored as files.
Shared OSID

Functional Description
The Shared OSID contains fundamental objects used in the other OSIDs to provide their functionality.

Why important
The contents of the Shared OSID are used throughout O.K.I.-compliant implementations and applications.

Use cases
A person, a requesting service, an external O.K.I. instance, or an O.K.I. tool are all examples of objects using the Agent interface.
Scheduling OSID

Functional Description
The Scheduling OSID manages events in shared calendars.

Why important
Class schedules are an example of events that are managed in shared calendars.

Use cases
An application could use Scheduling to allow students to interactively select their preferred class sections.
UserMessaging OSID

Functional Description
The UserMessaging OSID supports communication and notification among users.

Why important
Person to person (P2P) messaging has become a useful application feature with the availability of supporting P2P services as well as e-mail, instant chat, and discussion boards.

Use cases
A distributed learning system might want to provide a means for distributed students to interact with an instructor in real time.
Workflow

Functional Description
The Workflow OSID provides a way to manage an interdependent succession of activities each of which has completion constraints.

Why important
Certain types of applications have operations where one action is dependent on the completion of a previous action.

Use cases
Workflows are commonly used to manage approval processes.
Question & Answer
Resources

http://mit.edu/oki/