Logical Data Model

- Community
- Collection
- Item
- Bundle
- Bitstream
- Format
- DC Record

Diagram shows relationships between these entities, indicating the multiplicity of connections.
Metadata I

- **Descriptive:**
  - Qualified Dublin Core in RDBMS; registry of elements and qualifiers
  - Other schema (e.g. MARC) stored as serialised bitstreams

- **Administrative:**
  - Some stored as Dublin Core (e.g. provenance)
  - Mostly stored in legacy DB schema
• **Structural:**
  - Currently, just structure of data model
  - Proposed use of METS to describe relationship between bitstreams in bundles

• **Preservation:**
  - Registry of bitstream formats with preservation metadata
• Much of the metadata proprietary, embedded in RDBMS schema

• Defining Archival Information Package based on METS to store this in a standard way
Persistent Naming

• Handle system from CNRI
• Currently Handles assigned to items - soon to cover collections and communities
• CNRI Handle resolving software with DSpace “plug-in”
• Individual Handles managed within DSpace system
Implementation

- Implemented in Java
- UNIX-type operating systems: Linux, HPUX etc
- Open source – BSD-style license
- Uses only free, open source libraries
Storage Layer

• RDBMS: PostgreSQL via JDBC, and a simpler “homegrown” API on top

• Simple bitstream storage API - currently just stores bitstreams in the file system
Content Management API

- Java classes for manipulating content and metadata
- Fully transaction-safe
- Invokes authorisation system
- Records object history
Workflow System

• Submissions to a collection may go through editorial or review process

• Implemented simple submission workflow with three optional stages; meets initial requirements
Search

- Simple wrapper around Jakarta "Lucene" search engine
- Lucene maintains indices, implements stemming, stop-word removal and wildcard searching
- Wrapper allows bounded search within specific community/collection
Browse

• “Homegrown”

• Builds indices in RDBMS

• Browse by author, issue date, title

• Bounded browse within specific community/collection
“E-people” and Group Management

- Currently represent only users of system: Not a naming authority
- Stores user’s e-mail address, real name, password, contact information
- Simple group management – organise e-people into named groups
Authorisation

Policies are triples:
(object, action, group)

- Any object (community, collection, item, bitstream) can have access policies, and policies can be inherited from containers
History

- Mechanical history of objects stored in RDF - Harmony ABC model
- Stored in file system
- Simple index held in RDBMS
Ingest

• Legacy import format (SIP) defined - soon to be based on METS

• To ingest, content + metadata transferred to temporary filespace on the server

• Ingest application is invoked, which trawls and ingests data

• Ingested content can be sent through or bypass workflow system
Administration Toolkit

- Ubiquitous “Context” object is used for authorisation and transaction-safety
- Configuration system - includes handling of configuration files for external tools (e.g. Apache, Tomcat)
- Standard text-based logging
- Classes for manipulating Dublin Core and bitstream format registries
Web User Interface I

- Implemented as Servlets (for processing HTTP requests) and JSPs (for displaying HTML) - allows easy modification of display without affecting functionality

- Tomcat 4.0 Servlet engine running with Apache (for SSL features)

- Caucho Resin also supported
Web User Interface II

• Authentication with e-mail/password or X509 certificates

• Other sites can provide a site authenticator class with custom code, e.g. interfacing with a departmental database
Web User Interface III

Pages for:

• Search
• Item display

• Browse
• Item submission

• Configurable community and collection home pages

• “My DSpace” - managing in-progress submissions and workflow tasks

• Central administration UI - currently little support for self-administration by communities