The Use Of Message-Driven Workflows On The Service Bus Pattern for Indexing Fedora Repositories

Open Repositories 2011

Adam Soroka
Online Library Environment
the University of Virginia
Library
Motivations
Indexing is Curation
Indexing is Curation

Indexing metadata is metadata
Indexing is Curation

Indexing metadata is metadata

Keep indexing together with indexed material
Indexing is Curation

Indexing metadata is metadata

Keep indexing together with indexed material

Put indexing into the hands of curators
Indexing can be Continuous
Indexing can be Continuous

There is nothing natural about batch indexing
Indexing can be Continuous

There is nothing natural about batch indexing

Asynchronous workflows *can* have better scaling characteristics
Use the same configurations for data and workflow
Use the same configuration

Reduce maintenance
Use the same configuration

Reduce maintenance

Only works if repository-idiomatic tools are the right tools
Step 1: Make indexing asynchronous
Step 1: Make indexing asynchronous

Done.

Thanks, Gert!
Step 1: Make indexing asynchronous

Use JMS event streams
Step 1: Make indexing asynchronous

Use JMS event streams

Add Web service connectivity
Step 2: Bring index configuration into the repository
Step 2: Bring index configuration into the repository

Simple beginnings: only XML metadata
Step 2: Bring index configuration into the repository

Simple beginnings: only XML metadata

Create objects to represent configuration
Step 2: Bring index configuration into the repository

Simple beginnings: only XML metadata

Create objects to represent configuration
  • Content models (types)
Step 2: Bring index configuration into the repository

Simple beginnings: only XML metadata

Create objects to represent configuration

- Content models (types)
- Disseminations (behaviors)
Step 2: Bring index configuration into the repository

Simple beginnings: only XML metadata

Create objects to represent configuration
- Content models (types)
- Disseminations (behaviors)
- Use behaviors to hide state *only as desired*
Step 2: Bring index configuration into the repository

Simple beginnings: only XML metadata

Create objects to represent configuration

Keep indexing machinery outside repository
Object relationships
Step 2½: The machinery of indexing
Step 2½: The machinery of indexing

- Apache Camel
- Jetty
- Saxon
- Apache HttpClient
- Apache Velocity
- Apache ActiveMQ
- inter alia

- Apache ServiceMix: JBI container

- Apache Karaf: provisioning, configuration

- Apache Felix: OSGi container

- JVM

- OS
Step 2½: The machinery of indexing

Bus (routing, message transformation)

Service
Service
Service
Service
Service
Service
Within the Bus

Router: repository-connectivity
- Camel
- Velocity
- Saxon

Router: indexing
- Camel
- Velocity

Metadata Transform
- Saxon

Split RDF descriptions
- Saxon

RDF URI concretization
- Saxon

Merging and deduplicating
- Saxon
Router: Indexing

indexing:index

Is Indexable?

Yes

indexing:indexIndexableObject

http://fedora/getIndexingMetadata

To Solr

No

Is Indexer?

Yes

Next slides

No

Done

To Solr
Router: Indexing

indexing:getIndexMetadata

Retrieve datastreams and transformations

Transform

Transform

Transform

Merge and deduplicate
Step 3: The future
Step 3: The future

Indexing multiobject records (ECM Views)
Step 3: The future

Indexing multiobject records (ECM Views)

Indexing non-XML metadata
Step 3: The future

Indexing multiobject records (ECM Views)

Indexing non-XML metadata

Indexing RDF to external (non-RI) triplestores
Step 3: The future

- Source code available soon

- Virtual instance test drive available now
  - http://mbusdev.lib.virginia.edu/or2011/demo.ova.gz