

SSE3: Advanced Software Technologies for Knowledge Management

Professor Uffe Kock Wil
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark

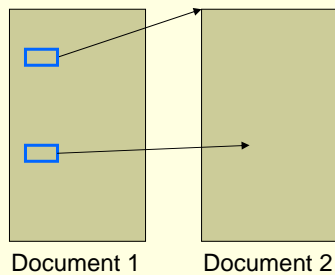
Lecture 3: Topics

- Issues
 - Open versus closed
 - Structure awareness / domains / abstractions
- Open hypermedia
 - 3. Levels of integration, integration models
 - 4. Architectural model for integration
 - 5. Multiple open services
- Class exercise
 - “Reverse engineer” an OHS integration

Open versus closed

■ Closed

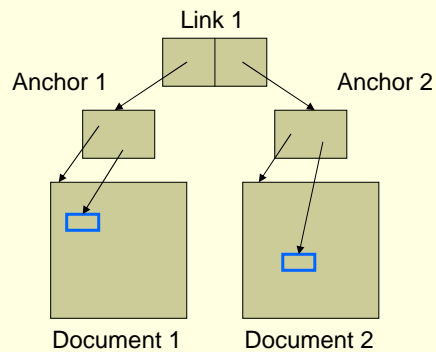
■ Example: web



SSE3 Spring 2010

■ Open

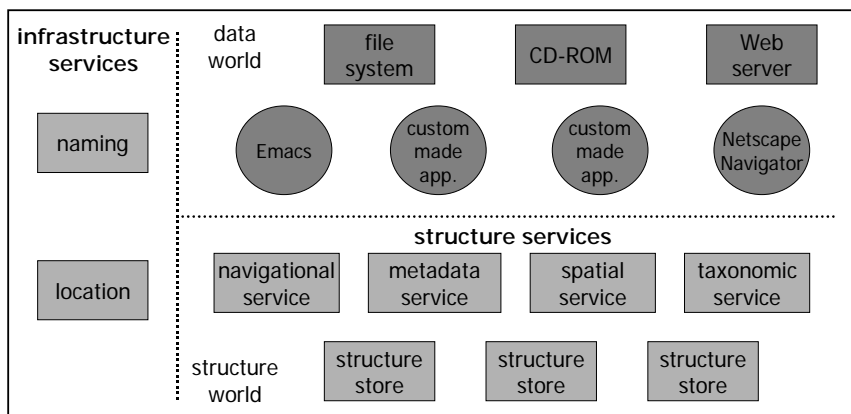
■ Example: ohs



Lecture 3

3

Data / structure separation



SSE3 Spring 2010

Lecture 3

4

Structure awareness

■ How deep does it go?

- Application
- Middleware
- Storage
- OS
- Hardware
- ...

Structure awareness



Human layer

Application

Application layer

Structure
service

Structure service layer

Storage

Structure storage layer

Structure domains

- Navigational (association)
- Spatial (information analysis)
- Taxonomic (classification)
- Issue-based (argumentation support)
- Annotation (annotation)
- ...

Exercise: Structure abstractions

- Navigational
 - (node, link, anchor, ...)
- Spatial
 - ??
- Taxonomic
 - ??
- Issue-based
 - ??
- Annotation
 - ??

Issues

- Assuming that we haven't seen the light – that we need for structuring mechanisms in our daily work with knowledge ☺
- No OS support for structuring available ☹
 - How can we build new (native) applications that support structuring mechanisms?
 - How do we introduce structuring mechanisms into existing (3rd party) applications?
 - Can we have one “framework” that supports both?

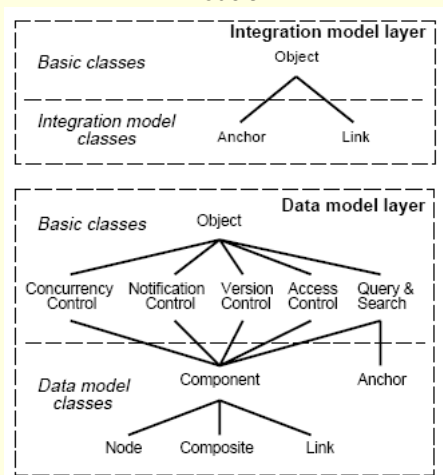
SSE3 Spring 2010

Lecture 3

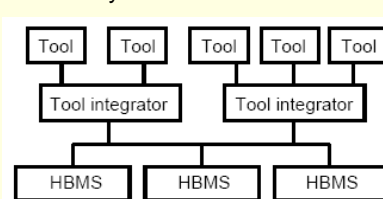
9

HyperDisco: Example OHS

Models



System architecture



HBMS = Workspace

10

Models

- Integration model layer
 - Basic hypermedia linking services (including a flexible communication protocol)
 - Encapsulate tool-dependent hypermedia linking functionality
 - Interface to general hypermedia functionality
 - Individually tailored integration models and communication protocols
- Data model layer
 - Basic hypermedia storage services
 - Creation, storage, retrieval, updates, queries and deletion of hypermedia objects

Workspace

- Workspace concept
 - Serves as a gateway to multimedia files
 - residing in the underlying file system
 - stored in the workspace itself
 - Provides storage and retrieval of file contents (nodes) and file handles (data wrapper nodes)
 - Offers a wide range of services to participating applications

Workspace services

- Storage and Integration
 - Individual storage models for integrated applications
 - Structure and contents storage (content nodes and file-wrapper nodes)
- Hypermedia
 - Anchors and Links
 - Composites
- Collaboration
 - Access control on files (nodes) and links
 - Locking
 - File monitoring (using events) & versioning of contents
- Distribution
 - Multiple distributed servers (workspaces): LAN / Internet
 - Access control: private, public & shared workspaces

Levels of integration

- Tools can be integrated at different levels
 - Full integration
 - Documents stored in different node types in HyperDisco
 - Links to and from documents
 - Anchors within documents (to pieces)
 - Partial integration
 - Like full integration, except that:
 - Only links, anchors and data-wrapper nodes stored in HyperDisco
 - Documents stored elsewhere
 - No integration (launch only)
 - Like partial integration, except that:
 - Only links to documents (not from)
 - Anchors to whole documents (not pieces)

Levels of integration

■ Levels of integration

■ Full integration



■ Partial integration



■ No integration (launch only)

■ Requirements on tools

■ Store documents in HyperDisco

■ Handle events from HyperDisco

■ Manage component OID's

■ Manage anchor tables

■ Manage communication

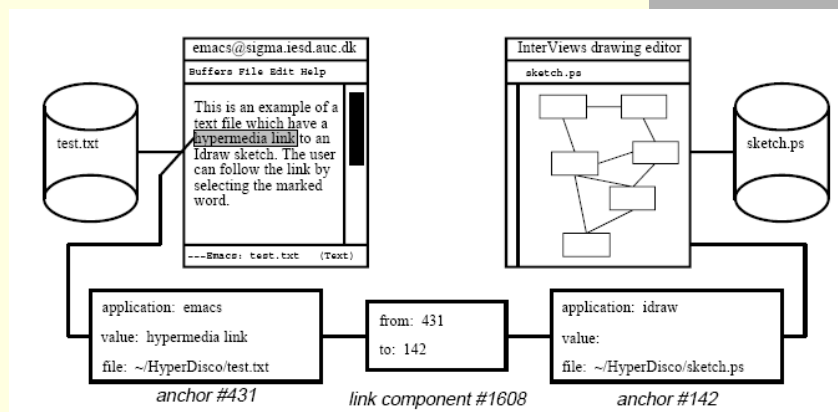
■ No

SSE3 Spring 2010

Lecture 3

15

Integration of tools



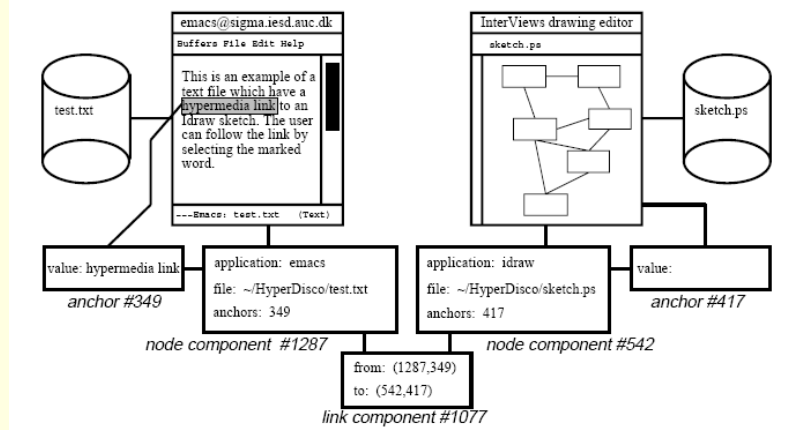
Anchor link integration model

SSE3 Spring 2010

Lecture 3

16

Integration and extension of tools



Data wrapper node integration model

SSE3 Spring 2010

Lecture 3

17

Integration method

- Integration tasks
 - 1. Analyze tool to determine the degree of openness
 - 2. Determine appropriate level and model of integration
 - 3. Specialize the built-in hypermedia services to support the chosen level and model of integration
 - A. Integration model
 - B. Data model
- Extension tasks
 - Tool specific

SSE3 Spring 2010

Lecture 3

18

Architectural model of application integration

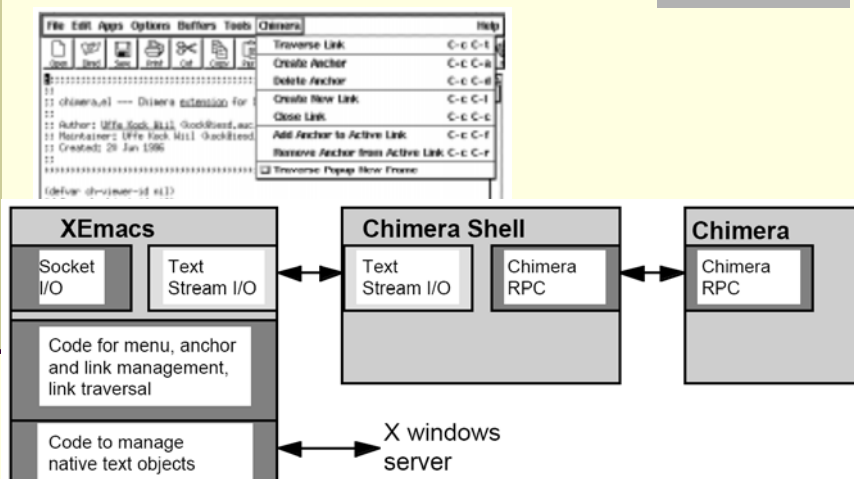
- Introduces a method for creating architectural models of application integrations with OHS
- The modeling framework can be used to
 - Model applications prior to integration
 - Describe the characteristics of a *complete* integration
 - Provide guidance in selecting the architecture of a finished integration
 - Provide a rough estimate of the degree of effort required to perform an integration
 - Categorize existing integrations
- Gathers community knowledge
 - Microcosm, DHM, HyperDisco, Chimera, etc.

SSE3 Spring 2010

Lecture 3

19

XEmacs/Chimera integration

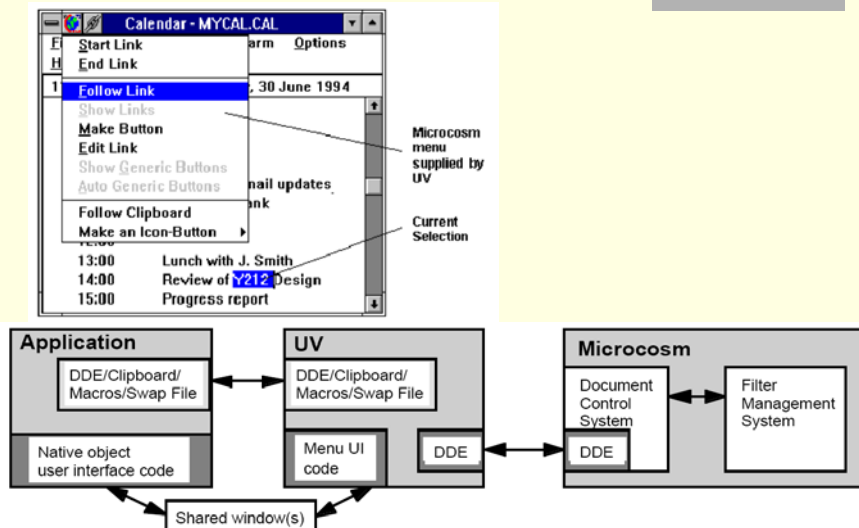


SSE3 Spring 2010

Lecture 3

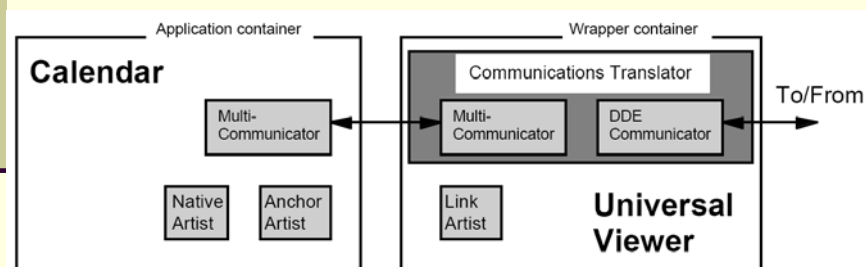
20

Calendar/Microcosm integration

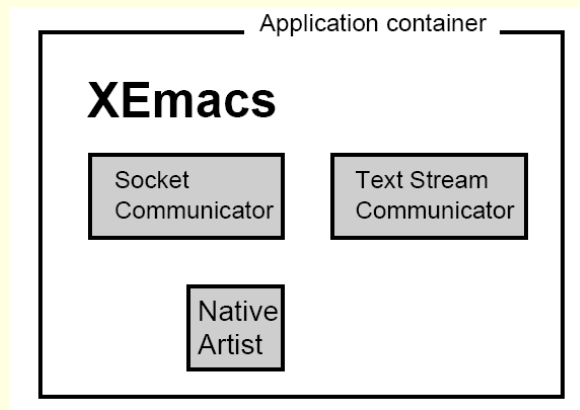


Architectural model of integrations

- Artists model UI aspects
- Containers group elements
- Communicators model communications



Application prior to integration



SSE3 Spring 2010

Lecture 3

23

Common integration architectures

- Launch only
 - Non-communicative application
- Wrapper
 - Intermediary component
- Custom
 - Modify source code or extend application using a built-in extension language
- Combination
 - Launch only + wrapper
 - Custom + wrapper

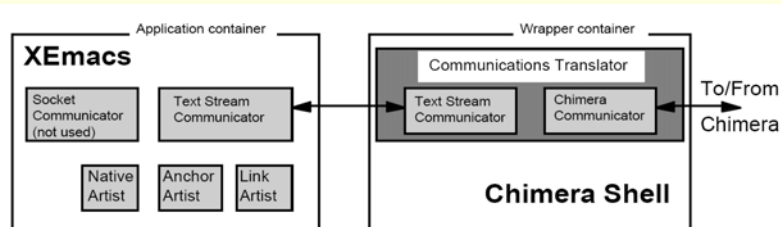
SSE3 Spring 2010

Lecture 3

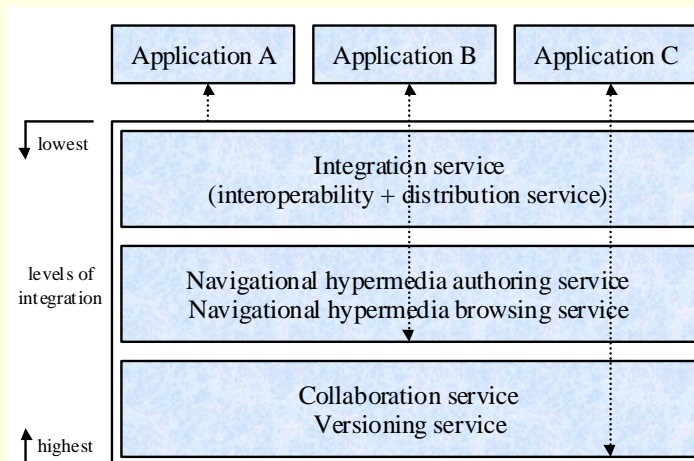
24

Properties of a complete integration

- Existence of all three artists: native, anchor, and link
- Existence of a communicator for the hypermedia system's native protocol
- If the application has a communicator for a protocol which is different from the hypermedia system's protocol, then a communications translator must be present which translates into the hypermedia system's protocol



Multiple Open Services Case Study: HyperDisco



Multiple Open Services Requirements

- 1. Open architectural framework ✓
 - CB-OHS
- 2. General availability ✓
 - Across computing platforms and OS
- 3. Open service
 - 3.1 Orthogonality of services
 - 3.2 Generality of services
 - 3.3 Scalability of services

SSE3 Spring 2010

Lecture 3

27

Multiple Open Services Requirements

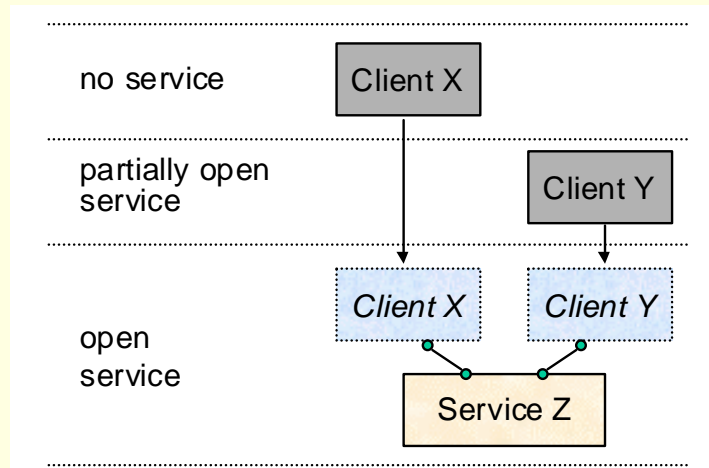
- 3.1 Orthogonality of services
 - Functional independence of other services
 - Should not overwrite existing services (UI, storage format, etc.)
 - No overlap to other services
- 3.2 Generality of services
 - Useful across applications
- 3.3 Scalability of services
 - Different level of services

SSE3 Spring 2010

Lecture 3

28

Multiple Open Services Open Service Provision

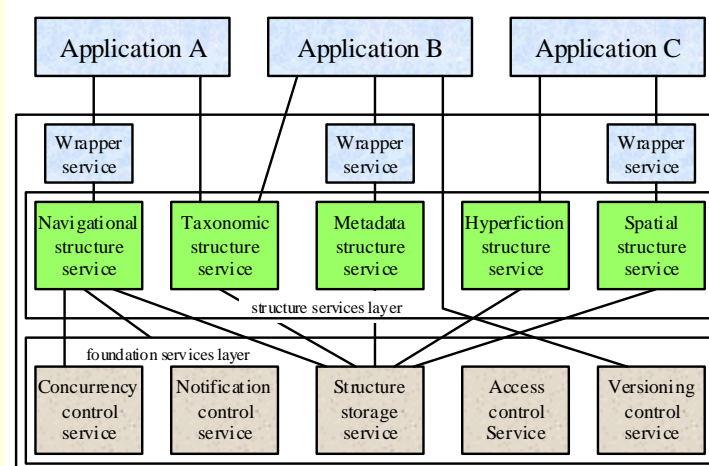


SSE3 Spring 2010

Lecture 3

29

Multiple Open Services Implications (1)



SSE3 Spring 2010

Lecture 3

30

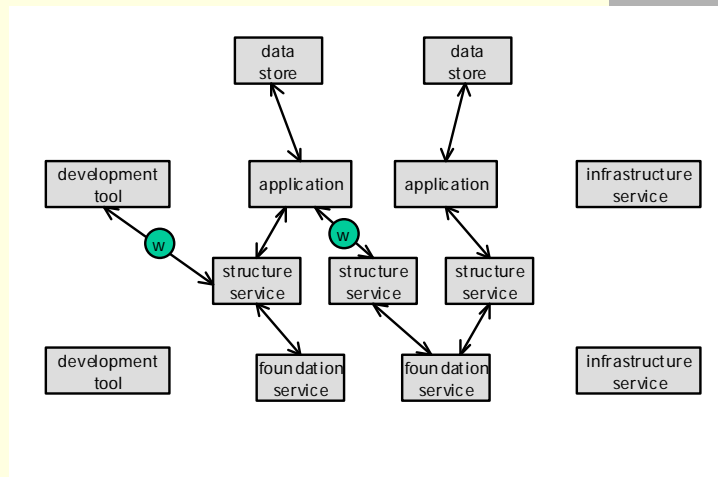
Multiple Open Services Implications (2)

- Services belong to a layer
- All services at a layer must comply with the requirements
 - Orthogonality, generality, scalability
- Each layer is open to new services
 - If the above is fulfilled
- Any component can in principle be a client of any other component

Multiple Open Services Advantages

- Towards a highly modular and flexible architectural framework
 - Standardization and interoperability of services at all levels (not just middleware services)
 - including foundation services
 - Service modularization (decoupling)
 - explore interactions between services more easily
 - services can be the focus of attention and research independently (to gain a deeper understanding)

Construct: System Overview

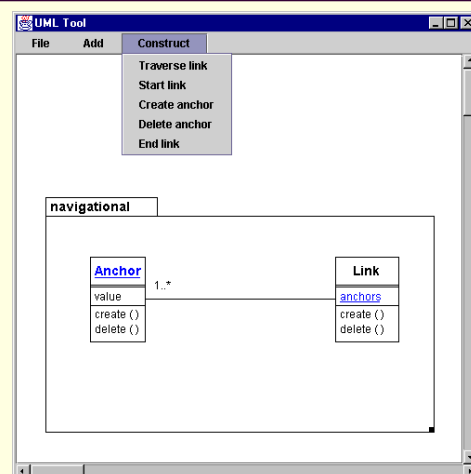


SSE3 Spring 2010

Lecture 3

33

Construct: Example Application Integrations (1)

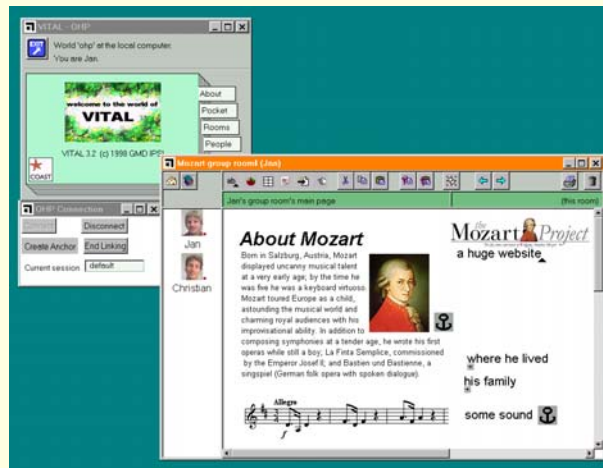


SSE3 Spring 2010

Lecture 3

34

Construct: Example Application Integrations (2)

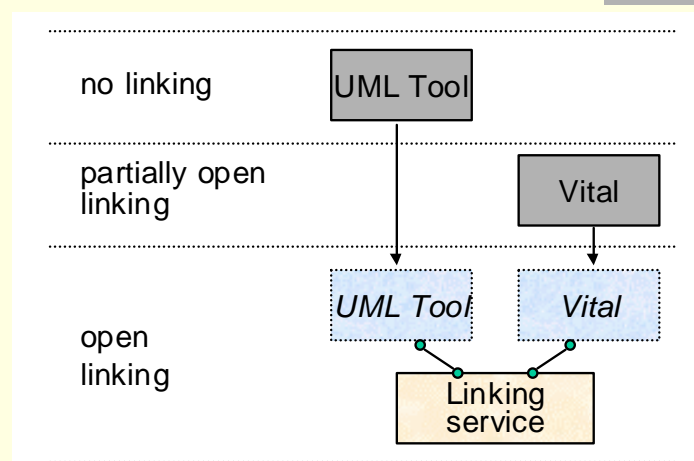


SSE3 Spring 2010

Lecture 3

35

Construct: Open Service Provision

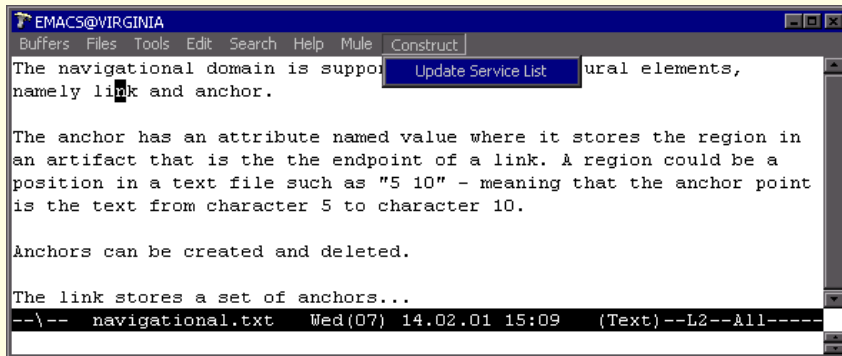


SSE3 Spring 2010

Lecture 3

36

Construct: Managing Multiple Open Services (1)

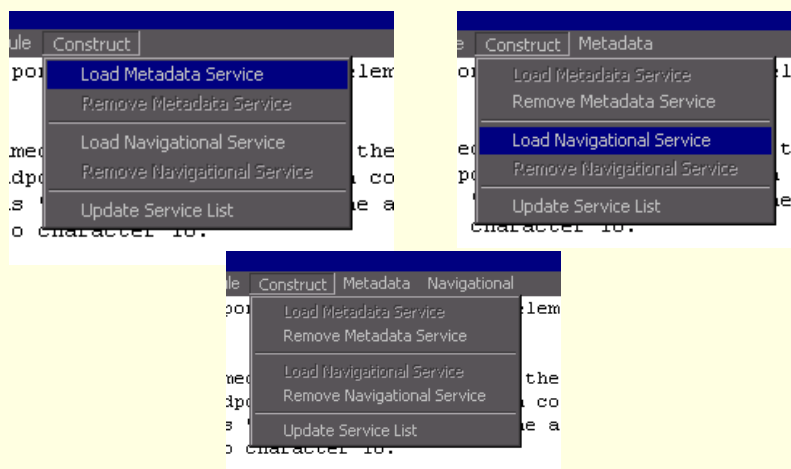


SSE3 Spring 2010

Lecture 3

37

Construct: Managing Multiple Open Services (2)

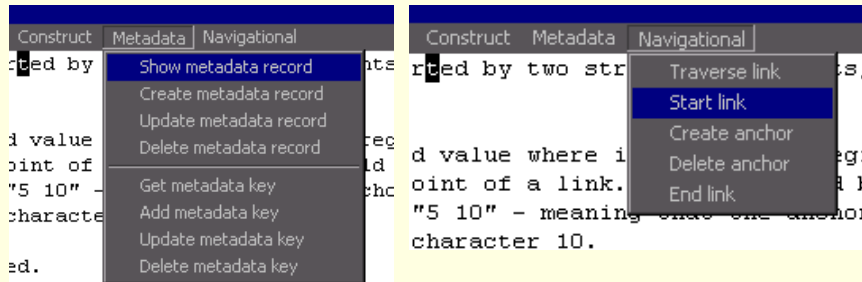


SSE3 Spring 2010

Lecture 3

38

Construct: Managing Multiple Open Services (3)

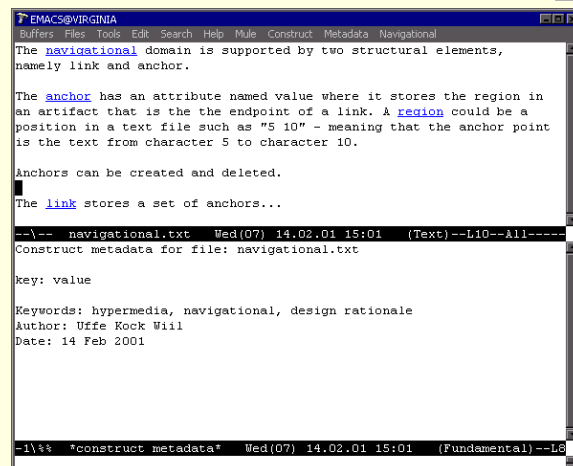


SSE3 Spring 2010

Lecture 3

39

Construct: Managing Multiple Open Services (4)



SSE3 Spring 2010

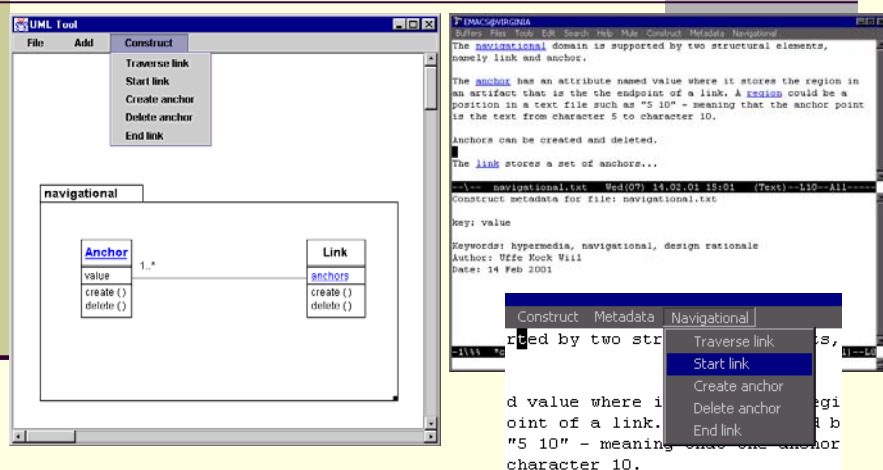
Lecture 3

40

Class exercise

- Reverse engineer the EMACS / UML Tool integrations with Construct (see next slide)
 - Architectural model?
 - Define API's

Navigational structures



Questions

- The three papers (3 + 4 + 5)

Exercise

- Design an integration of a known application to another hypertext structure domain (other than navigational)
 - Application integration
 - Wrapper API
 - Structure service API