SSE3: Advanced Software Technologies for Knowledge Management

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

Lecture 4: Topics

- CSCW
  - Introduction
  - Definition
  - Basic concepts
- Hypertext and CSCW
  - Concepts
  - Mechanisms
  - Example systems
- Exercise

CSCW concepts

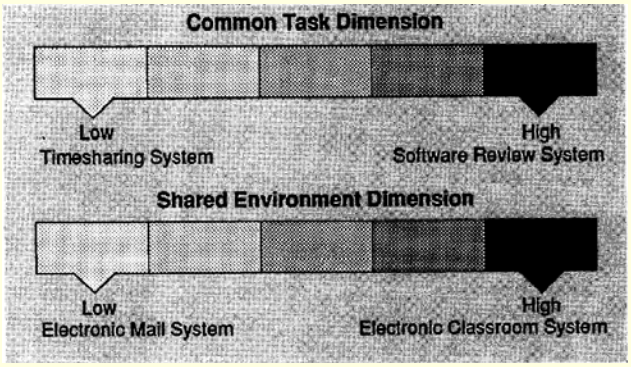
- Computer Supported Cooperative Work
  - Computer systems deal with human computer interaction
  - There is also need for dealing with human human interaction
- Two types of CSCW (groupware)
  - Real-time
  - Non-real-time

CSCW concepts

- 3 central concepts for group interaction
  - Communication
    - Real world: Face-to-face and phone
    - Computer: Email, video/audio conference, ….
  - Collaboration
    - Sharing of information
    - Problem: databases isolate users
  - Coordination
    - Conflicts must be handled
    - Introduces a necessary overhead

CSCW definition

- Common task (or goal)
- Shared environment



CSCW taxonomy

	Same Time	Different Times
Same Place	face-to-face interaction	asynchronous interaction
Different Places	synchronous distributed interaction	asynchronous distributed interaction

CSCW perspectives

- Distributed systems
  - Decentralization of data and control
- Communications
  - Exchange of information between remote agents
- Human computer interaction
  - User interface in computer systems
- Artificial intelligence
  - Techniques and technologies for enhancing machines with human like attributes
- Social theory
  - Sociology (human behavior)

Real-time concepts

- Shared context
- Group window
- Telepointer
- View
- Synchronous and asynchronous interaction
- Session
- Role

- Example: GROVE multiuser editor

CSCW issues

- Group processes
  - Support for communication, collaboration, coordination
- User interface (for groups)
  - WYSIWIS
  - Dynamical aspects
  - Distraction
  - Response time

CSCW issues

- Concurrency control
  - Data sharing, including:
  - Replication, centralization/decentralization, serialization of actions, transactions, locking (granularity, who / what initiates)
- Access control
  - Granularity
- Event notification
  - Granularity

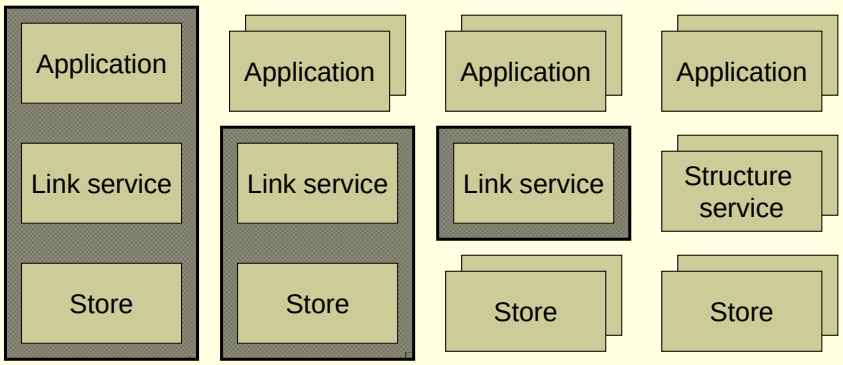
CSCW issues

- Different modes of cooperation
  - Different levels of awareness
- Smooth transitions between modes
  - A single author manipulates some data
    - Individual mode
    - (asynchronous collaboration)
  - A new user arrives
    - Loosely-coupled mode
    - (increased awareness of other users)
  - A session is started; both users enroll in the same session
    - Tightly-coupled mode
    - (synchronous collaboration)

Hypertext and CSCW

- Example hypertext systems that support CSCW in different steps of the architectural evolution
  - KMS step 1
  - EHTS step 2
  - SEPIA step 2
  - HyperDisco step 3
  - Construct step 4
- Summary

Architectural evolution

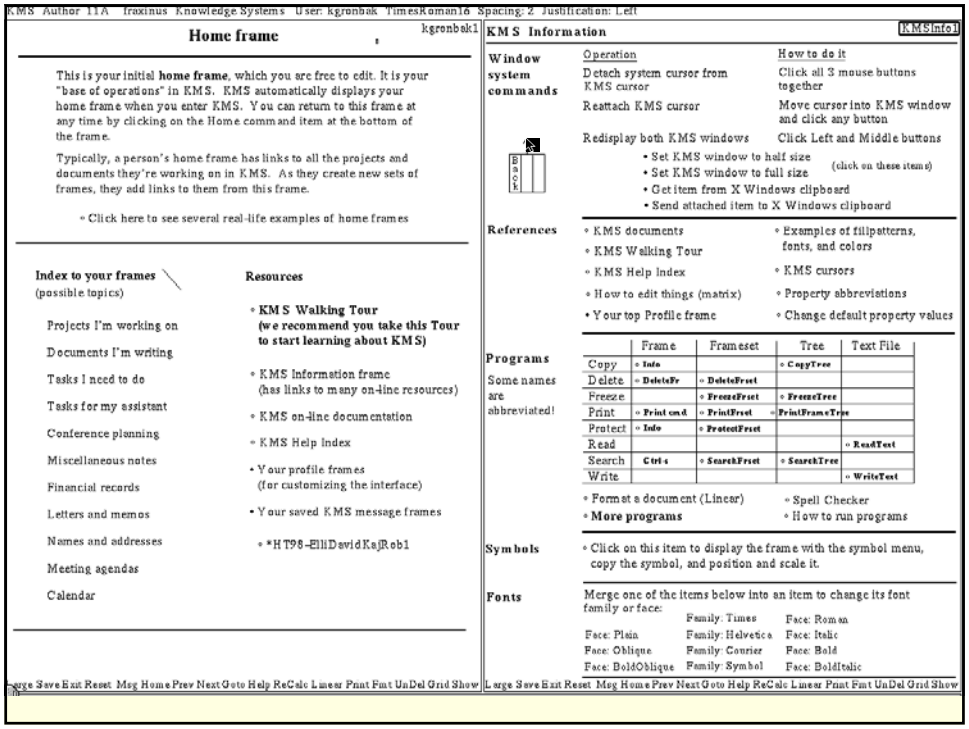


Monolithic system

Client-server system

Open hypermedia system (OHS)

Component-based OHS (CB-OHS)

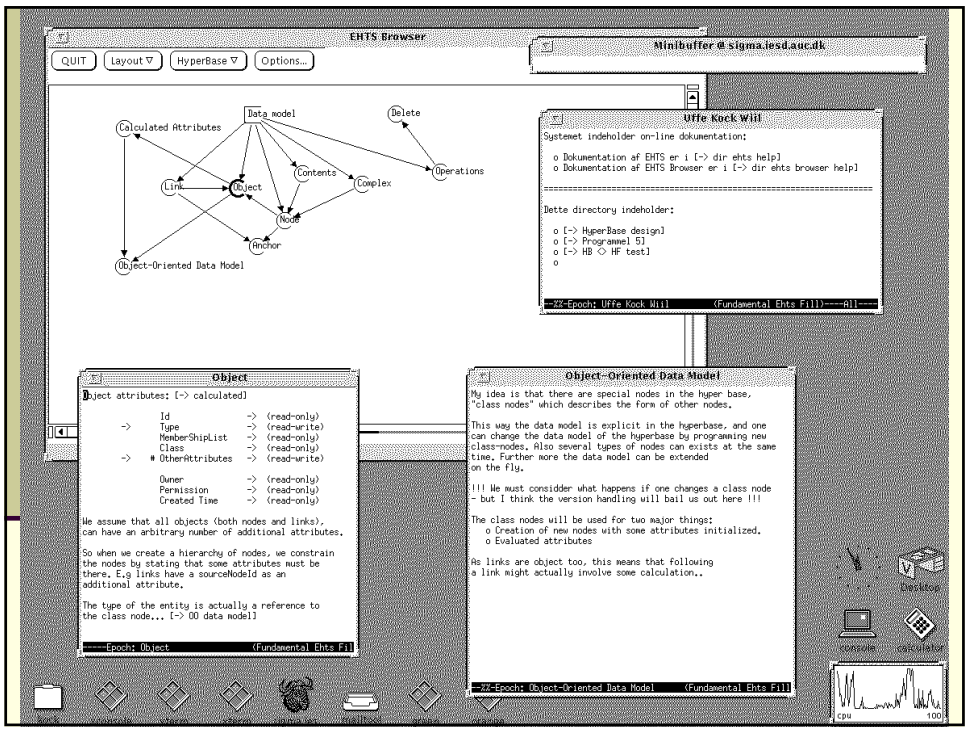


Uffe K. Wiil – a former KMS user

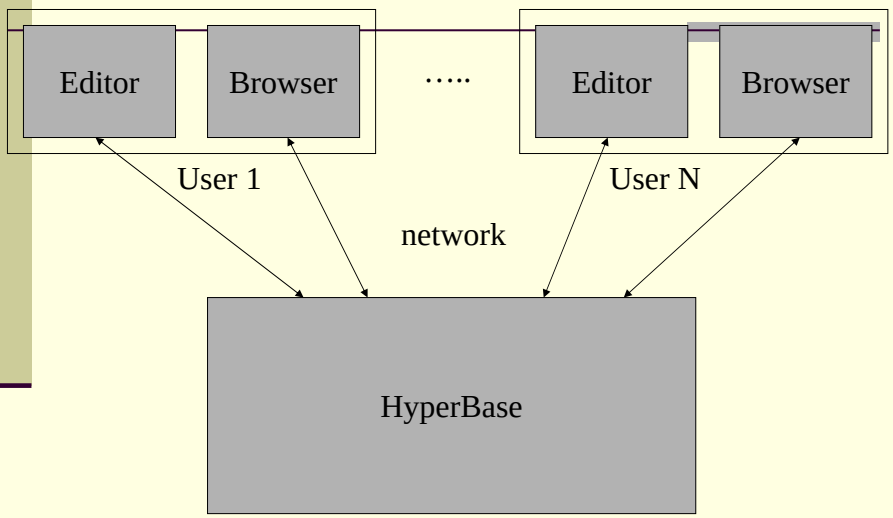
- Experienced KMS user
  - 1991-2004
  - Many papers, notes, slides, letters, designs, etc.
- Had a personal KMS database of about 500.000 frames
- The KMS installation in Esbjerg had 3 “real” users
  - Shared framesets for common work
- Collaboration with John J. Leggett i Texas via KMS
  - Email integrated with KMS via action language
  - A frameset is wrapped and send in an email
  - Unwrapped and installed
  - Several papers written in this way
  - Allows annotations in margin
  - Allows links to comments
  - Allows links to old versions

CSCW & KMS

- Basic principals
  - Based on “relaxed groups norms”
  - Ad-hoc communication, collaboration and coordination
- User interface
  - Relaxed WYSIWIS, no dynamics and distraction, good response time
- Concurrency control
  - Multiple users can edit the same frameset (even same frame)
  - Optimistic concurrency control to avoid locking
  - In case of a conflict between updates on the same frame, a new temporary copy is made for subsequent updates
- Access control
  - Frame owner can protect frames against modification
  - Annotations by others can be allowed (annotation items not visible when printed)
- No event notification


EHTS Browser
QUIT
Layout
HyperBase
Options...
Minibuffer @ sigma.iesd.auc.dk
Data model
Calculated Attributes
Delete
Operations
Contents
Complex
Link
Object
Node
Anchor
Object-Oriented Data Model
Uffe Kock Wiil
Systemet indeholder on-line dokumentation:
o Dokumentation af EHTS er i [-> dir ehts help]
o Dokumentation af EHTS Browser er i [-> dir ehts browser help]
Dette directory indeholder:
o [-> HyperBase design]
o [-> Programmel 5]
o [-> HB <> HF test]
Object
Object attributes: [-> calculated]
Id -> (read-only)
Type -> (read-write)
MemberShipList -> (read-only)
Class -> (read-only)
OtherAttributes -> (read-write)
Owner -> (read-only)
Permission -> (read-only)
Created Time -> (read-only)
We assume that all objects (both nodes and links), can have an arbitrary number of additional attributes.
So when we create a hierarchy of nodes, we constrain the nodes by stating that some attributes must be there. E.g links have a sourceNodeId as an additional attribute.
The type of the entity is actually a reference to the class node... [-> OO data model]
Object-Oriented Data Model
My idea is that there are special nodes in the hyper base, "class nodes" which describes the form of other nodes.
This way the data model is explicit in the hyperbase, and one can change the data model of the hyperbase by programming new class-nodes. Also several types of nodes can exists at the same time. Further more the data model can be extended on the fly.
!!! We must considder what happens if one changes a class node - but I think the version handling will bail us out here !!!
The class nodes will be used for two major things:
o Creation of new nodes with some attributes initialized.
o Evaluated attributes
As links are object too, this means that following a link might actually involve some calculation..
Desktop
console
calculator
cpu

CSCW & EHTS


Editor
Browser
.....
Editor
Browser
User 1
User N
network
HyperBase

CSCW & EHTS

- General hypertext-based multiuser editing system
- System components
  - Client-server system
  - Database server, text editor client, graphical browser client
- User interface
  - Relaxed WYSIWIS
    - Placement of windows
    - Browser layout
    - Update of shared information
  - Dynamics can result in distraction
  - God response time
- Communication, collaboration and coordination
  - Real-time communication ~ communication
  - Real-time monitoring ~ collaboration
  - Access contentions ~ coordination

CSCW & EHTS

- Concurrency control
  - Nodes can be locked for individual updates
  - Other users can se the content but cannot alter it (dirty reads)
- Event notification
  - Specific actions in the database trigger events, which cause explicit actions
  - Users are informed about changes made by other users
  - Automatic updates of the shared user interface
- Access control
  - Access to editor can restricted
  - Graphical browser cannot alter the information in the databasen

CSCW & EHTS

- Real-time communication (communication)
  - Talk system (user name)
  - Message system (user name / all users)
- Real-time monitoring (collaboration)
  - Editor
    - Node content (including new links)
    - Node attributes (name, font, window size)
  - Graphical browser
    - Lock and unlock
    - Changes to hypertext structure, inclusing:
    - Creation / deletion of nodes
    - Creation / deletion of links
    - New endpoints for links

CSCW & EHTS

- Access contentions (coordination)
  - Event (name) when locked (if node open in a window)
  - Event (name) if locked node is opened in a window
  - Event (name) when attempting to locke already locked node
- Possible actions in EHTS as a response to a conflict
  - Context the “locking party” (talk or message)
  - Subscribe to unlock event
  - Keep window open and follow the updates to the node
  - Try again later

CSCW & SEPIA

- Very different application domains, but in many ways based on the same solutions as EHTS
  - locking
  - events
  - client-server model
- The most important different is that SEPIA supports smooth transition between different modes of cooperation
  - individual mode: working on separate parts of the shared (handled by the database)
  - loosely-coupled mode: “awareness”, managing conflicts and coordination
  - tightly-coupled mode: synchronous cooperation, shared environment

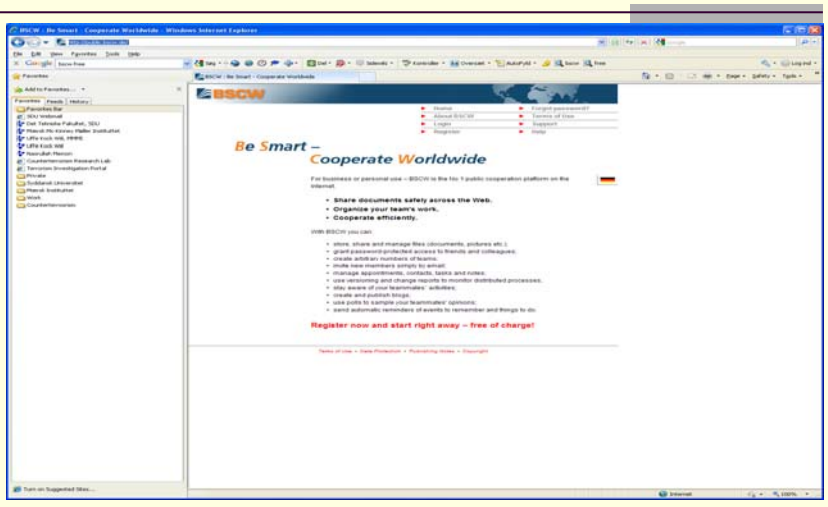
EHTS and SEPIA experiences

- Identified several mechanisms that are central for CSCW
  - User-controlled locking
    - Shared locking
    - Fine-grained locking
    - Persistent locking
  - Event notification
    - Fine-grained notification
    - Persistent notification
  - Transaction management
    - Short transactions
    - Long transactions
  - Version control
    - Removes the need to lock – every user has own version
    - Maintains document history – who has changed what when
    - Concurrent updates to same document – merge later
    - Adds complexity

CSCW & OHS

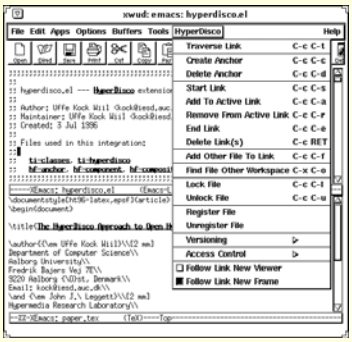
- Evolution of hypermedia systems
  - Closed -> open
  - Local area network -> Internet
- Questions:
  - Are the well-known techniques applicable to OHS?
  - Do they scale to the Internet (the web)?
  - Do we need new models for collaboration on the Internet?
- BSCW (http://public.bscw.de)
- HyperDisco

BSCW



Presentation of Distributed Workspaces

XEmacs as a participating HyperDisco application

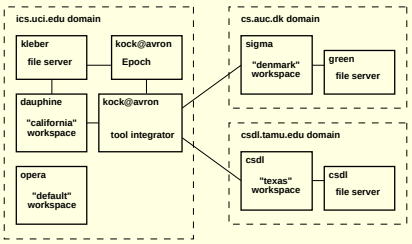


Anchors are bold-faced and underlined. HyperDisco operations are available from the keyboard and through the menu. Links can also be followed by double-clicking on the anchors.

- **workspaces <-> buffers**
- **link creation scenario**
- **link traversal (frame / viewer)**
- **add endpoints in other files**
- **multiple views (app & file)**
- **load from any workspace**

Internet Distribution of Workspaces

Single user distributed workspace scenario



User "leggett" can join by:

- **starting tool integrator on his personal workstation**
- **starting his favorite integrated editing tool**

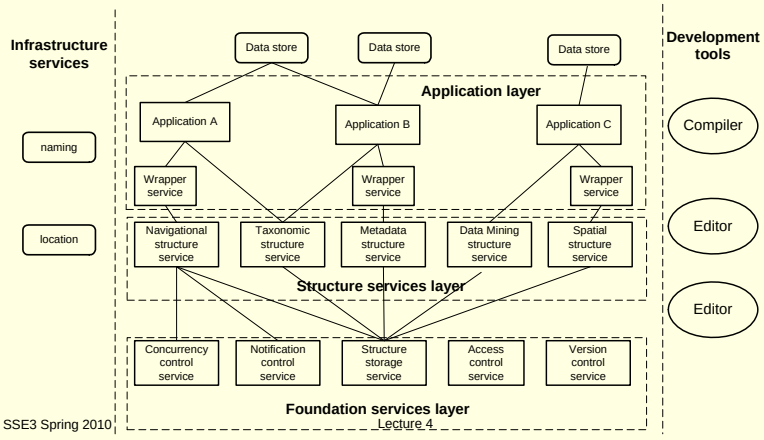
"get" and "put" files
name service

An example scenario with four workspaces ("default", "california", "denmark" and "texas") running at different Internet domains

Cooperation services in Construct

- Example CB-OHS
- Scenario
- Implications
  - New services

Conceptual architecture



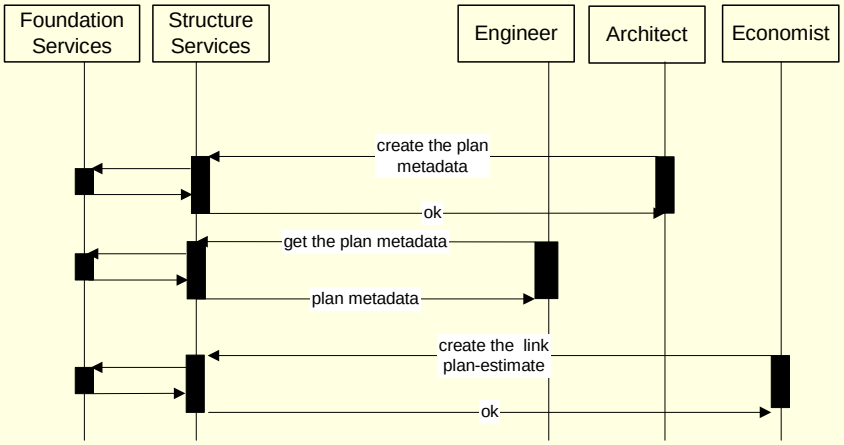
Work scenario

- Assume that in a given architectural design project there are three cooperating partners: an architect, an engineer, and an economist.
- The architect is the person responsible for the production of the final plan of the building and all of the related documents.
- However, the engineer can be in charge of some parts of this plan or of some technical details.
- All along the design process, the economist produces cost estimations corresponding to the current version of the plan. The architect reads these cost estimations in order to check the compatibility of the project with the budget conditions.

Work scenario

- *First, the architect opens and updates the plan using his favorite application. After that, he creates metadata using the Construct metadata service for the current plan that will later be used by the engineer to perform her work. This metadata may represent some rules (specifying some technique choices or the sketch of the plan) established by the architect (the person responsible for the plan design) that will be followed by the engineer in her work.*
- *Later, during the process of the plan design, the economist needs to produce estimates of the building construction cost corresponding to the current plan. For this reason, she creates a new estimate using her favorite application. Then, she creates a link between the new estimate and the current plan. She uses the Construct navigational structure service for this.*

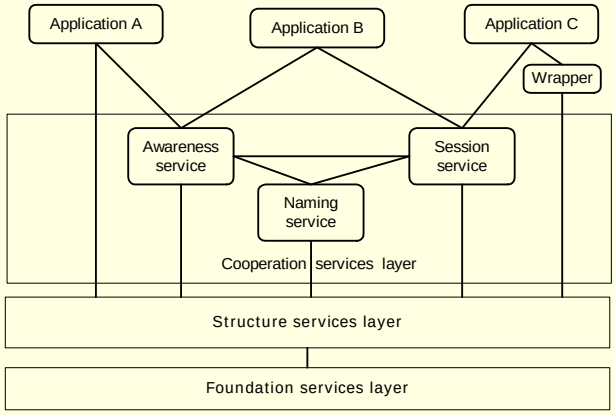
Work scenario



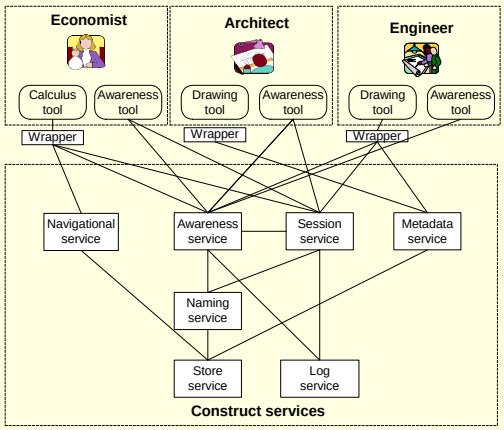
Implications

- Need for new services
  - Awareness
    - Modification of shared material
  - Session
    - Architect – economist (group 1)
    - Architect – engineer (group 2)
    - Groups: different relationships and goals
- Requirements on existing services
  - Naming service
    - Associate name with actual component

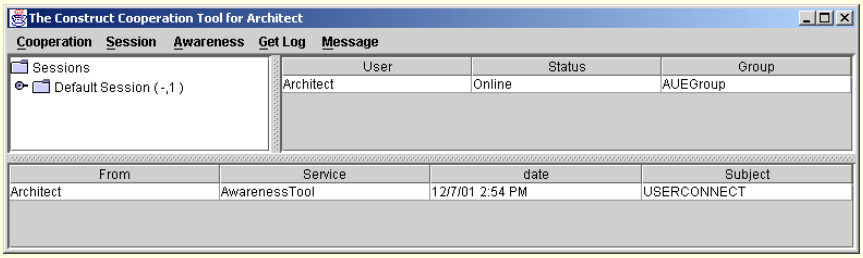
New conceptual architecture



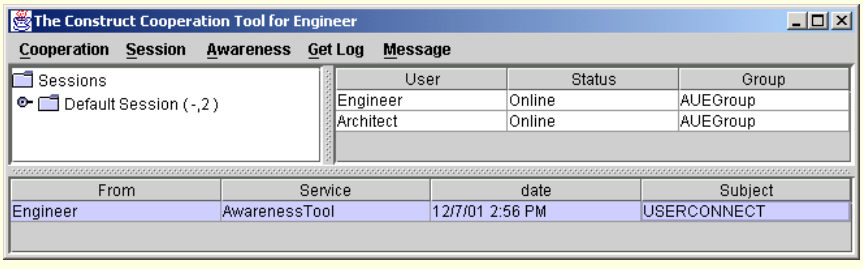
Tools and services in the scenario



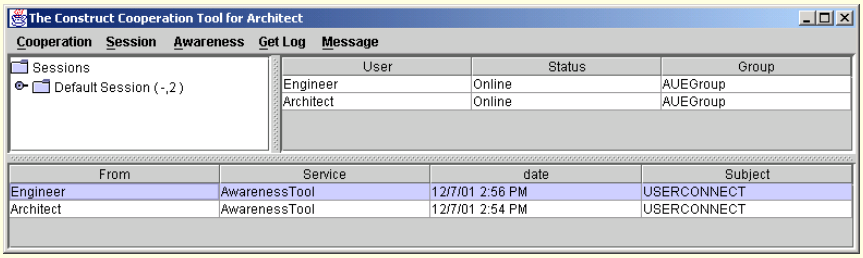
Scenario – step 1



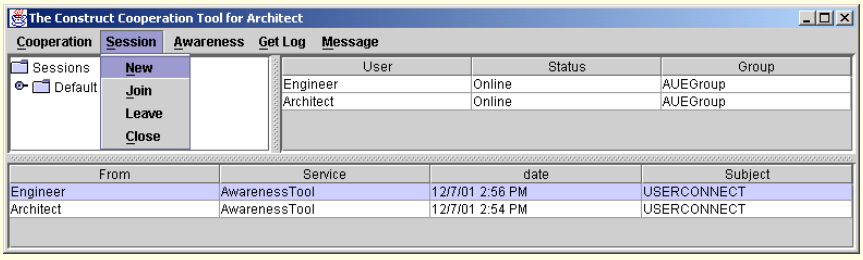
Scenario – step 2



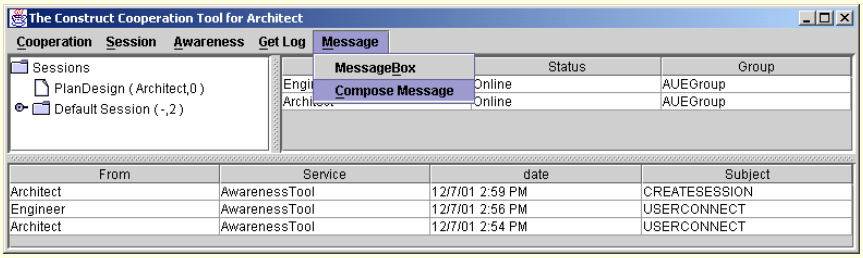
Scenario – step 3



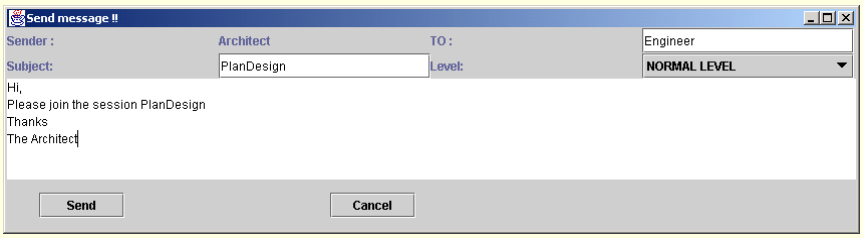
Scenario – step 4



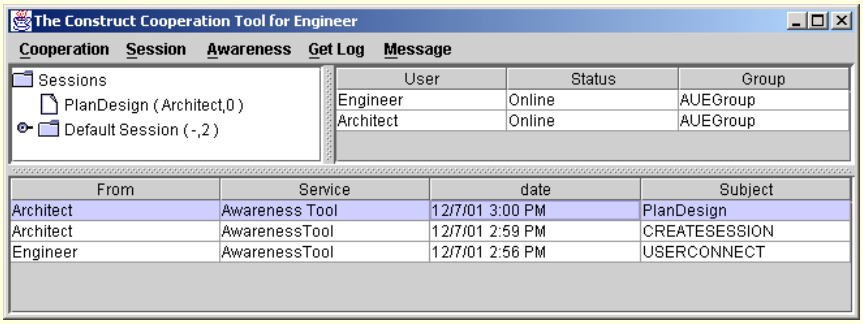
Scenario – step 5



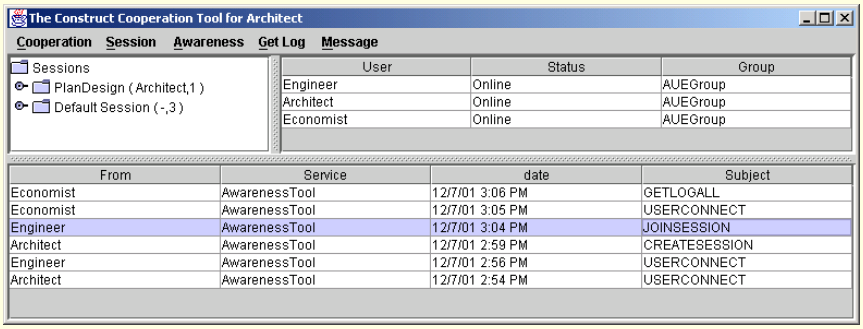
Scenario – step 6



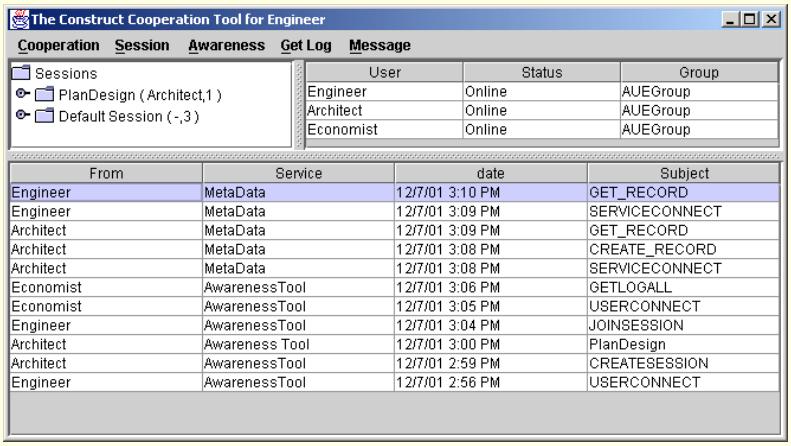
Scenario – step 7



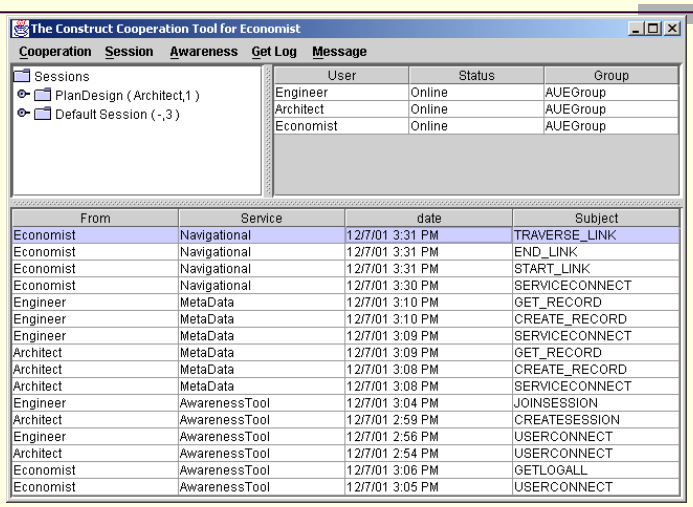
Scenario – step 8



Scenario – step 9



Scenario – step 10



Questions

- The three papers (6 + 7 + 8)

Exercise

- Think about the concepts and processes of CSCW and the mechanisms supporting CSCW
  - What concepts and processes were introduced today?
  - What underlying mechanisms are needed to support them?