LINKÖPING UNIVERSITY
IDA/IISLAB
Martin Sjölin

LINCKS – ett verktyg för CSCW

Martin Sjölin

April 25, 1994

LINCKS

LINCKS är ett fleranvändar klient-server system som kontrukterad ovanpå en objekt centrerad databas (NODE)[1]. LINCKS har ett rikt stöd för dynamiska "vyer" (multipla), temporal datahantering, sammansatta objekt, versionhantering samt CSCW tillämpningar.

Struktur

LINCKS är en databas (ej distribuerad) där klienter på arbetsstationerna är uppkopplade gentemot "servern" via ett lokalt nätverk. Systemet kan grovt delas upp i följande delar:

1. NODE (databasen), en server som består av tre olika (UNIX) processer för att: validera användare, skapa unika objektidentiteter, och att lagra och hämta objekt.
2. NODE klienter vilka tillsammans med (1) bildar ett system för att hantera objekt.
3. "Composition Management", ovanpå (2), filtrar objekten, bygger upp och hanterar de sammansatta objekten och vyerna.
4. Gränssnittet, vilket hanterar interaktionen gentemot användaren. Idag finns ett hypertext liknande X11 snitt med "emacs" kommandon för att skapa och ändra textdokument.

Objekt & Versioner

Ett objekt i databasen har i princip tre olika delar:

1. Egenskaper (strängar) som i någon mening ger objektet en typ[2],
2. Länkar till andra objekt. En länk kan antigen peka till en specifik *version*, en så kallad bunden länk, eller den är obunden, dvs den refererar till den senaste versionen av objektet.

[1] Vi använder termen "objekt centrerad" då meddelande hantering inte finns mellan objekten.
[2] Alla objekt i den bakomliggande databasen är otypade. Det är tillämpning i sig som avgör typerna.

3. Binär data – i den nuvarande X11 tillämpningen är det text, även om postscript, filer, bilder, eller annan data information är möjligt.

Versioner är ett viktigt concept i LINCKS - i databasen sparas *alla* versioner vilka representerar tillstånden för objekten under deras utveckling. I de fall att användare förändrar och/eller sparar objekt samtidigt så kan parallella versioner utvecklas.

CSCW

Genom att använda (parallella) versioner istället för att låsa ett eller flera objekt medans en användare arbetar så kan en medarbetare fortsätta sitt eget arbete. En sammanslagning av objekten och deras innehåll måste givetvis göras senare, men arbetet kan fortgå. Denna (potentiella) konflikt måste i *alla* fall lösas via någon typ av överenskommelse mellan de inblandade parterna.

I de fall att användare A påbörjar en förändring av objekt Q (denna paragraf) och sedan hämtar användare B denna paragraf, så kommer användare B att erhålla ett meddelande om att användare A redan arbetar med paragraf Q. Om nu B börjar ändra i paragraf Q erhåller användare A ett meddelande om att B har börjat ändra i Q. Konflikten får användare A och B lösa via en sekundär kommunikationskanal (telefon, TV, mail, LINCKS etc).