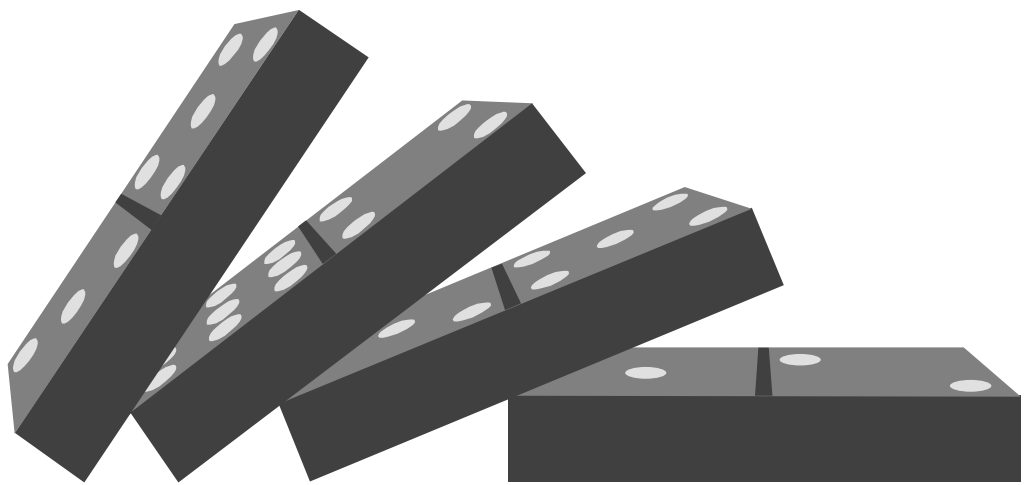


Allt faller i bitar



Komponentbaserad systemutveckling och upphovsrätt

INNEHÅLLSFÖRTECKNING

1	INLEDNING	3
1.1	VAD ÄR KOMPONENTER?	3
1.2	KOMPONENTER OCH DATORPROGRAM.....	4
2	HUR SKYDDAS DATORPROGRAM?.....	5
2.1	UPPHOVSRÄTT.....	5
2.2	PATENT	8
3	NYA FRÅGESTÄLLNINGAR.....	10
3.1	AVGRÄNSNINGSPROBLEM	10
3.2	GEMENSAM UPPHOVSRÄTT	13
3.3	ÖVRIGA FRÅGOR	15
3.3.1	<i>Komponenter och skadestånd</i>	<i>16</i>
3.3.2	<i>Komponenter och agenter.....</i>	<i>16</i>
3.3.3	<i>Komponenter och bevisning.....</i>	<i>17</i>
3.3.4	<i>Komponenter och global handel.....</i>	<i>17</i>
3.3.5	<i>Reverse engineering och dekompilering.....</i>	<i>18</i>
4	EN MARKNAD FÖR KOMPONENTER?	19
4.1	MODELLER	19
5	SLUTSATSER.....	22
6	LITTERATUR.....	23

1 INLEDNING

I detta arbete skall några frågeställningar om en speciell typ av datorprogram belysas. I SISU-publikation 98:03 *Informations sambällets juridik* förutskickades att SISU skulle granska moderna företeelser inom informations- och kommunikationsteknologin närmare, ur ett rättsvetenskapligt perspektiv. Komponentbaserad systemutveckling är vad som kallas ett "hett ämne" och det erbjuder juristen några intressanta och inte helt triviala problem som kan vara värdefulla att bena ut.

Arbetet har disponerats så att vi först kommer att presentera idén med komponenter helt kort, för att därefter jämföra datorprogram och komponenter. Därefter följer ett avsnitt som förklarar hur datorprogram skyddas idag. I det tredje kapitlet berörs de nya frågor som uppstår när vi hanterar komponenter. Avslutningsvis försöker jag skissa en modell för hur en marknad för komponenter skulle kunna se ut.

1.1 VAD ÄR KOMPONENTER?

Det ges inte något självklart svar på frågan om vad komponenter är. Som ordet antyder rör det sig om delar som kan sättas samman till helheter. Vi kan *komponera* system med komponenter, för att ta ett aktuellt exempel.

Ofta kan komponenter ses som små program som medvetet konstruerats för att kunna sammanfogas med andra komponenter. Anledningen till att använda komponenter är då att en ofta förekommande funktion inte behöver skrivas om var gång den skall ingå i ett system. Systemutvecklaren kan istället ta den färdiga komponenten och sedan lägga till andra komponenter som fyller hans eller hennes behov.

EXEMPEL 1

Per utvecklar ekonomisystem för medelstora företag. I ett uppdrag har Per fått uppgiften att sätta samman ett system som kan sköta fakturering, lagerhållning och kundregister. Utifrån tidigare arbete har Per redan stommen till dessa delar, han har komponenter som utför dessa uppgifter och som dessutom kan sammanfogas ganska enkelt. Genom att använda dessa komponenter undviker Per att varje gång tvingas skriva nya program.

Komponenter kan vara av mycket olika storlek. Från de komponenter som nämns i exemplet som modellerar delar av en affärsverksamhet till mycket små komponenter som kanske bara utför en enkel statistisk analys. Komponenter kan också vara delar av varandra. En komponent för fakturering kan till exempel innehålla färdiga komponenter för att sända meddelanden via Internet.

Denna inledning har med nödvändighet varit icketeknisk. Det viktiga från en jurists perspektiv är att komponenterna är *små datorprogram*. Som vi kommer att se är detta det centrala.

1.2 KOMPONENTER OCH DATORPROGRAM

Efter att ha sagt att komponenter är små datorprogram är det nödvändigt att också skissera skillnaderna mellan ”vanliga” datorprogram och komponenter.

Den största skillnaden ligger egentligen i hanteringen. Vanliga datorprogram är oftast redan sammansatta och färdiga. Tänk på ett ordbehandlingsprogram. Det är vare sig nödvändigt eller vanligt att vi kombinerar detta med något annat program på eget bevåg. Det kan förstås ske att mjukvaruleverantören levererar en integrerad lösning, ett office-paket till exempel, men själv förväntas jag knappast snickra ihop nya kombinationer av ordbehandlingsprogram och kalkylprogram.

Komponenter, å andra sidan, är gjorda för att sammanfogas till helheter. Ofta förutsätts att jag fogar samman komponenterna till något för att de alls skall bli funktionella. Jag kanske köper en komponent för stavningskontroll, en för utskrift och en för textredigering. Om jag nu vill skapa ett ordbehandlingsprogram måste jag foga samman dessa delar.

Denna skillnad gör att de juridiska förutsättningarna förändras radikalt. I det första fallet är jag användare. Jag manipulerar inte programmets funktionalitet annat än genom tillägg och uppdateringar. Licensavtalet kanske inte ens tillåter mig att förändra programmet. I det andra fallet är jag fortfarande användare, men jag tar också aktiv del i sammanfogandet av den slutliga produkten. Jag blir utvecklare i någon mening och alltså upphovsman.

Det är denna aspekt som gör att de lagar och avtal som styr vanliga datorprogram inte räcker till för att hantera den framväxande komponentmarknaden.

Även om de lagar och det skydd som idag finns för datorprogram inte kan fullständigt hantera den nya situation som komponentbaserad utveckling skapar, så kan de åtminstone ge viss ledning. Vi skall nu se kort på hur datorprogram skyddas idag.

2 HUR SKYDDAS DATORPROGRAM?

Vi har redan konstaterat att komponenter i rättsvetenskaplig mening är datorprogram. Det kan därför vara värdefullt att närmare studera hur skyddet för datorprogram är uppbyggt idag. När vi talar om skyddet av datorprogram kommer vi osvickligen in på upphovsrättens område, och i ett vidare perspektiv, immaterialrätten.

Immaterialrätten är den del av juridiken som reglerar rätten till uppfinningar, litterära och konstnärliga verk, varumärken och mycket annat som vi inte kan ”ta på”. Här ingår upphovsrätt, varumärkesrätt, patenträtt, mönsterrätt och många andra delområden som kommit att betyda allt mer i det framväxande informationssamhället.

Det är framförallt ett delområde, upphovsrätten, som kommer att stå i fokus här. Skälen till detta är enkla: upphovsrätten reglerar lejonparten av skyddet för datorprogram.

2.1 UPPHOVS RÄTT

Datorprogram skyddas i första hand enligt Lag (1960:729) om upphovsrätt till litterära och konstnärliga verk (URL). Redan i första paragrafen återfinns vi datorprogrammen:

1 § Den som skapat ett litterärt eller konstnärligt verk har upphovsrätt till verket oavsett om det är

- 1) skönlitterär eller beskrivande framställning i skrift eller tal
- 2) datorprogram [...]

Det känns kanske litet märkligt att skydda ett program som en konstnärlig eller litterär skapelse, men det finns goda skäl för regleringen. Det främsta är att skyddet omedelbart uppkommer utan någon speciell anmälan eller registrering. Om man valt exempelvis patent skulle ett helt annat ansökningsförfarande fordrats.

Samtidigt har det påpekats att datorprogrammen är något av en gökunge i upphovsrättens bo, eftersom de har stor ekonomisk betydelse. Den fokusering på immaterialrättsliga frågor som kännetecknat mycket av de senaste årens rättsvetenskapliga utveckling har delvis sin förklaring i upphovsrättens växande roll i informationssamhället.

Hur skyddas ett datorprogram i Sverige idag? Vi skall gå igenom hela processen och se kort på hur lagen griper in i olika stadier och vad som skyddas. För att få någon struktur har jag valt att åskådliggöra problematiken med ett konkret exempel. Vi kommer att följa Per från design av ett datorprogram till färdig produkt i olika exempel.

EXEMPEL 2

Per förbereder arbetet med att konstruera ett dataspel. Han ritar flödesdiagram, skissar på hur karaktärerna ser ut och skriver en liten intrig för att kunna modellera spelet. Materialet är ganska detaljerat och originellt. Är detta material skyddat på något sätt? Det har ju inte blivit något program av det ännu?

Förberedande designmaterial är skyddat (URL 1 § 3 st). Vad förberedande designmaterial är finns inte fastslaget. Flödesscheman, grafiska framställningar m.m. kan antas vara sådant designmaterial (Westman, Lindberg s.77) Pers ansträngningar vinner därmed skydd samtidigt som materialet skapas *om det har den grad av originalitet som krävs*.

Detta är centralt. Ett verk måste uppnå vad som ibland kallas ”verkshöjd” för att vara skyddat enligt upphovsrättsliga regler. Detta innebär att verket måste vara originellt och nytt. Se på följande exempel:

EXEMPEL 3

Per skriver ett program som summerar två tal. När han ser att en kollega har skrivit samma program hävdar Per att det är upphovsrättsintrång. Är det riktigt?

Svaret på denna fråga är nej. Om verket är trivialt eller enkelt är det inte skyddat. Detta är ett större problem än det i förstone ser ut att vara. Kravet på självständighet och originalitet ställs lågt i svensk rätt. Detta innebär att det skydd som erhålls inte blir alltför omfattande. Det krävs ganska litet skillnad mellan program för att de skall bedömas vara två fristående program.

Det finns ett helt annat och mer komplicerat problem att ta ställning till, som ansluter till denna problematik:

EXEMPEL 4

Per skriver ett program som utför en avancerad matematisk beräkning. Programmet gör endast detta och intet annat. Ingen annan har skapat ett program som utför denna beräkning förut och beräkningen i sig är att se som en matematisk landvinning. Är den skyddad enligt upphovsrättslagen?

Svaret här är mycket osäkert. Om programmet är det *enda sättet att utföra beräkningen på*, så kan det mycket väl vara så att det inte har skydd enligt upphovsrättslagen. Detta låter naturligtvis märkligt, men beror på att det inte *i lagens mening* behöver vara originellt att finna en ny matematisk formel. Ingen kan göra anspråk på upphovsrätten till integralkalkylen eller additionen. Nyutvecklad matematik är inte heller skyddad.

Detta kan verka märkligt. Litet av förklaringen finner vi i det faktum att upphovsrätten inte skyddar idéer, som vi kanske tror, utan realiseringar och konkretiseringar av idéer. Det är en idé *i en viss form* som skyddas. Inte idén själv. Om en matematisk beräkning bara går att utföra på ett sätt kan den alltså inte skyddas. Det är för mycket idé och för litet form, skulle vi kunna säga.

Låt oss gå vidare. När produkten är färdig träder alltså skyddet ikraft med en gång. Vad innehåller skyddet då?

EXEMPEL 5

Per har slutfört arbetet med sitt datorprogram och bränt en cd med det färdiga programmet. Denna har han lagt på bordet vid sin arbetsplats. Erik, Pers kollega, hittar cd:n och installerar spelet. Han blir så imponerad att han kopierar cd:n till sig själv och lägger ut spelet på sin hemsida på Internet. Per blir rasande, och vill stämma Erik – har det skett något oegentligt?

Den här beskrivna situationen är extrem, men har pedagogiskt värde. Ofta anses att upphovsrätten har dubbelt innehåll: den ekonomiska rätten och den ideella rätten. Den ideella rätten omfattar upphovsmannens rätt att bli nämnd och innehåller dessutom ett förbud mot kränkande ändringar av ett verk. När det gäller datorprogram har den inte så stor betydelse, eftersom en ändringsrätt ofta följer av licensavtal och liknande. Paternitetsrätten – rätten att omnämnas som upphovsman – kan dock ha viss ekonomisk betydelse för en programmerares renommé.

Den ekonomiska rätten, i sin tur, sönderfaller i två delar: rätten att mångfaldiga ett verk och rätten att göra ett verk tillgängligt. När vi skall bedöma Eriks beteende är det här vi finner ledning. Erik har utfört tre handlingar: han har a) installerat spelet b) kopierat cd-skivan och c) gjort verket tillgängligt på Internet. Samtliga dessa handlingar kränker Pers upphovsrätt. Låt oss gå igenom dem en och en.

a) *Installation*

Att installera ett program innebär tekniskt att vi gör en kopia av det. När vi köper ett program köper vi rätten att installera det på den egna datorn, rätten att göra en, eller vid licensavtal flera, kopior.

Den vanligaste missuppfattningen vi möter här är att det är tillåtet att kopiera för *personligt bruk*. Visserligen finns det en rätt att kopiera för enskilt bruk inskriven i URL 2 kap 12§, men denna omfattar uttryckligen inte datorprogram (URL 2 kap 12 § 2 st 2 p).

Den rätt som enligt 26 § g URL låter rättighetsinnehavaren göra en säkerhetskopior om det krävs för programmets avsedda användning bör inte sammanblandas med kopiering för enskilt bruk.

Eftersom mångfaldiganderätten tillkommer Per har Erik här gjort intrång i Pers upphovsrätt.

b) *Kopierandet av cd-skivan*

Att kopiera en cd-skiva är lika mycket kopierande som att installera. Rättsligt bedöms dessa situationer inte olika. Vad vi däremot kan anmärka här är att om någon kopierar ett stort antal cd-skivor så har detta betydelse vid utdömmande av skadestånd. Mer om detta nedan.

c) *Lägga upp verket på Internet*

Att lägga upp verket på Internet är egentligen inte att ”mångfaldiga” i lagens mening (en kopia måste i och för sig göras). Kopiorna görs ju av de som besöker

websiten. Men det är däremot att göra verket tillgängligt. Således är även detta intrång i Pers upphovsrätt.

Vad sker nu? Vi har funnit att Erik gjort intrång i Pers upphovsrätt i tre fall. Vad kan Per göra nu? I det enklaste fallet kan han helt enkelt stämma för upphovsrättsligt intrång och få ut en summa pengar. Ersättningen kommer att beräknas efter hur grovt intrånget bedöms vara. Hur allvarligt intrånget är bedöms utifrån flera olika faktorer, bl a utifrån hur många kopior som gjorts. En speciell svårighet kommer att uppstå under punkten c). Hur vet vi hur många kopior som folk har gjort av spelet under det att det låg ute på Internet? Här är det säkrast att Erik kan presentera någon typ av bevis, som loggfiler för sin website, annars kommer nog domstolen att gå på Pers uppskattning av hur många kopior som har gjorts.

Det skydd som Per åtnjuter på detta sätt löper under 75 år efter hans död.

Avslutningsvis är bör anställningsförhållanden nämnas.

EXEMPEL 6

Antag att Per har byggt spelet under sin anställning som datortekniker på spelbolaget. När spelet är klart ser Per att det verkar bli ett riktigt bra spel, och han beslutar sig för att starta eget. Han tar med sig spelet och börjar sälja det till olika återförsäljare. Efter en vecka blir han mycket förvånad när han får ett brev av sin forna arbetsgivare där han anklagas för upphovsrättsintrång. Det är ju han som gjort spelet! Han har inte ens skrivit på något avtal om att de skulle få se resultatet!

Per har fel. Det är spelföretaget som har upphovsrätten till datorprogrammet. Det framgår klart av URL 40 a §. Om inte annat avtalats övergår upphovsrätten till program framställda i anställningsförhållande till arbetsgivaren.

Här har i några, kortfattade och mycket enkla, exempel upphovsrättens grunder gått igenom. Det finns många andra intressanta frågor att dyka ned i, men fokus för den här rapporten är hur upphovsrätten hanterar komponentprogramvara. Innan vi kan fördjupa oss i den frågan måste vi se på möjligheterna att få patent på ett datorprogram i Sverige idag.

2.2 PATENT

Patent är ett skydd för uppfinningar. Det kan tyckas naturligt att låta datorprogram omfattas av detta skydd. Om ett datorprogram är någonting är det väl snarare en uppfinning än ett litterärt verk? Från lagstiftarens sida har man emellertid bedömt saken annorlunda. I 1 § Patentlagen (1967:837) står uttryckligen att

Såsom uppfinning anses aldrig vad som utgör enbart

- 1) en upptäckt, vetenskaplig teori eller matematisk metod,
- 2) en konstnärlig skapelse,
- 3) en plan, regel eller metod för intellektuell verksamhet, för spel eller för affärsverksamhet eller ett datorprogram,
- 4) ett framläggande av information.

Av punkten 3 framgår alltså att ett datorprogram ensamt aldrig kan omfattas av patent. Detta var tills alldeles nyligen en ganska enkel och hård princip. I ett

rättsfall från 1990 ändrades emellertid inställningen till datorprogram och patent. Det är komplicerat att sammanfatta ändringen enkelt, men i all enkelhet kan vi säga att om ett datorprogram är en del av en teknisk/industriell process så kan det omfattas av hela processens patent. Rättsfallet handlade om en programvara som igenkände och använde talsignaler i en process för att styra maskiner. (RÅ 1990 ref 84, jfr Westman, Lindberg s 122f)

Det anses allmänt att detta rättsfall inledde en period där det kan bli lättare att få patent på datorprogram. Denna ändring av patenträtten står i god samklang med de harmoniseringsåtgärder som vidtagits inom Europeiska Unionen, vars European Patent Office har visat liknande tendenser.

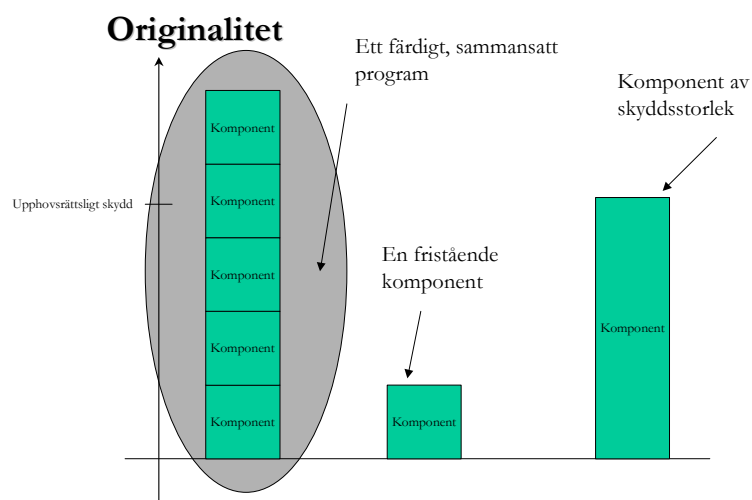
3 NYA FRÅGESTÄLLNINGAR

Eftersom komponenter rättsligt är att betrakta som datorprogram har vi inlett vårt studium med en kortfattad redogörelse för hur datorprogram skyddas. Det andra steget är att se hur komponenter skyddas, och vilka nya frågeställningar som aktualiseras av komponentbaserad systemutveckling. Två frågeställningar har urskiljt sig som mer betydelsefulla än de övriga och de är framförallt frågan om **avgränsningsproblem** och frågan om **gemensam upphovsrätt**. Dessa två frågor skall undersökas i detalj. Därefter kan vi kortfattat se på några enklare frågor.

3.1 AVGRÄNSNINGSPROBLEM

Ett vanligt datorprogram, som Microsoft Word 97, består av en mängd små moduler, men säljs som en enda produkt. Datorprogrammet i sin helhet anses ha den originalitet och individualitet som krävs för upphovsrättsligt skydd, även om en ordlista som ingår som separat modul inte har det. Om delarna har det eller ej är inte intressant, eftersom *allt säljs som en enda produkt*. Vad händer om vi bryter loss delarna och säljer dem var och en för sig?

En av de stora fördelarna med att arbeta med komponenter är att vi kan finna våra egna lösningar genom att sammanfoga olika moduler. Det upphovsrättsliga skyddet för dessa moduler försvinner dock i det ögonblick de blir så små att de blir triviala. Om det upphovsrättsliga skyddet försvinner kan produkten kopieras fritt och vilken tillverkare är intresserad av att skapa moduler som inte är skyddade? Om de saknar skydd blir det svårare för honom att ta betalt för dem. I värsta fall kan vi här se en utveckling där komponenter aldrig blir mindre än att de uppnår upphovsrättsligt skydd. Låt oss åskådliggöra detta i en bild.



Vad bilden försöker åskådliggöra är att en enskild komponent mycket väl kan vara så trivial att den inte får upphovsrättsligt skydd, samtidigt som en samling av komponenter har det. Konsekvensen av detta kan bli att komponenter endast tillverkas i sådan storlek att de erhåller skydd – eller litet större, som bilden visar.

Kan då inte mindre komponenter vinna skydd på något annat sätt? Ett ofta använt sätt är att **avtala om skydd för produkter** som egentligen kanske inte har rätt till upphovsrättsligt skydd. Det finns inget som hindrar att två parter kommer överens om att behandla ett föremål som om det hade upphovsrättsligt skydd. Skälet till att förfara på detta sätt kan vara att leverantören av komponenter kan leverera så många och så driftsäkra komponenter att köparen tycker att det är värt att betala för även de som inte egentligen är originella.

Om vi tänker oss att en leverantör säljer en samling enkla komponenter i ett bibliotek, uppstår en annan situation, eftersom det finns en paragraf, 49 §, i URL som skyddar sammanställningar. Skyddet är avsett som ett investeringskydd för den tid och de pengar som kan gå åt vid framställandet av en katalog eller sammanställning (Olsson, s.261ff). En samling av enkla produkter som var och en knappast är skyddad kan alltså bli skyddad *som sammanställning*. Värt att notera är att en sammanställning givetvis kan vara så originell att den vinner valigt upphovsrättsligt skydd.

EXEMPEL 7

Per sammanställer ett bibliotek med några favoritkomponenter för matematiska beräkningar som han har utvecklat. Erik tycker att det är toppen och börjar sprida det till alla han känner. När Per blir arg, påpekar Erik att de enskilda komponenterna bara utför enkla funktioner som medelvärdesberäkningar och normalfördelningar. Dessa skulle aldrig i sig få skydd. Per menar att den tid det tagit honom att sammanställa alla dessa komponenter till en fungerande helhet måste vara värd någonting, vem har rätt?

Per har rätt. Sammanställningen har skydd enligt 49 § URL. Det här skyddet är ganska svagt om vi jämför med vanligt upphovsrättsligt skydd, men det kan vara av viss ekonomisk betydelse. Svagheten ligger i att de enskilda komponenterna som ingår i sammanställningen inte är skyddade. Skyddstiden löper över 15 år från framställandet.

EXEMPEL 8

Erik bryter loss enskilda komponenter och skickar dem vidare till folk som tycks behöva dem. Per blir återigen irriterad och söker upp Erik för att få honom att sluta. Erik hävdar den här gången att de enskilda komponenterna inte är skyddade. Vem har rätt?

Eriks synpunkt är den riktiga. Lösa delar i sammanställningen får inte del av sammanställningens skydd. Här kan vi ana ett nyansproblem. Vad händer om Erik skickar två, tre, fyra...komponenter till någon bekant? Var går gränsen för en sammanställning? Finns det tidsaspekter här? Vad händer om Erik sänder dem tre och tre med några sekunders mellanrum? Ofta är det *praktiskt* inte svårt att bedöma detta, men problemet återkommer i ny form i nästa avsnitt.

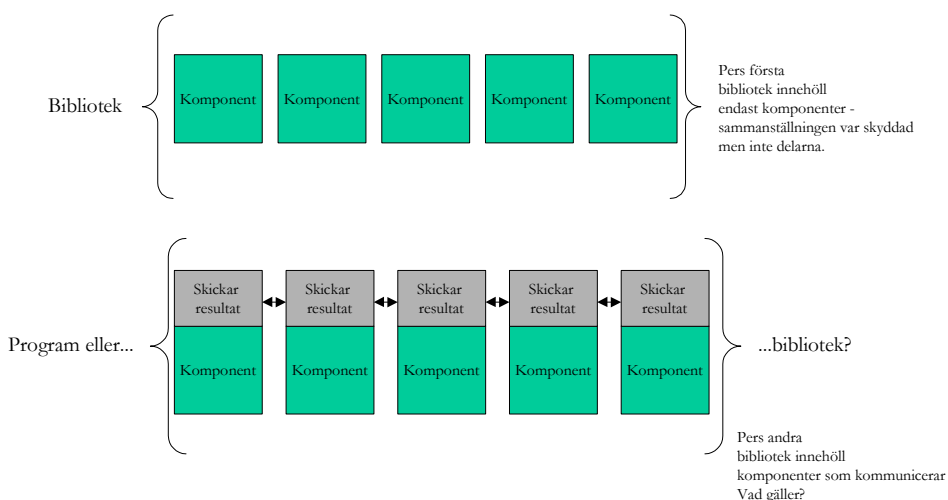
Ett mer sofistikerat problem uppstår om vi tänker oss följande situation.

EXEMPEL 9

Per har skapat ett bibliotek med en serie komponenter för matematiska beräkningar. Varje komponent fungerar helt fristående, men Per har i alla komponenter lagt in en funktion som gör att en komponent kan skicka resultatet av en beräkning till en annan komponent. Erik sprider enskilda komponenter, som i exemplet ovan. Per söker upp Erik och frågar varför Erik sprider delar av Pers **program**. Erik hävdar att det alls inte rör sig om något program, men Per är envis och säger att delarna samverkar och agerar tillsammans precis som i ett program. Erik invänder att varje enskild komponent fungerar ensam. Vad gäller?

Den här frågan är svårare att besvara än den tidigare. Enklast är att sammanställa problemet i en bild.

Vart går gränsen mellan bibliotek och program?



Det finns inget enkelt sätt att fastställa när en samling komponenter blir ett program eller en större komponent. Ett sätt vore emellertid att säga att om en komponent är fullt funktionell på egen hand så är den att betrakta som en självständig enhet som antingen omfattas av upphovsrättsligt skydd eller ej. Problemet är emellertid inte löst i domstol.

Kärnan i den sist betraktade frågan är en gränsdragningsproblematik. Klassisk programvara hade ganska tydliga gränser. Det skall dock inte förnekas att det även här fanns avsevärda problem.

EXEMPEL 10

Novell distribuerade en ny lösenordskomponent till Windows NT som tog över och bytte ut NT:s lösenordsfunktioner mot Novells när Novellnätverk användes. Fråga uppkom om Microsoft som stod bakom Windows NT hade någon supportskyldighet för de som valde att använda Novells produkter. Operativsystemet hade förändrats och var ej längre den produkt som kunden betalat för. Vad händer med produkten när en ingående del byts ut med en annan?

Svaret på denna fråga är ej heller det klart. Novell och Microsoft kom till någon sorts förlikning i ärendet och det fick bero. Var Microsofts produkt slutade och Novells började gjordes aldrig klart.

När vi utvecklar system helt och hållet baserade på komponenter blir problemet likartat. Var ”slutar” en komponent? I exemplet ovan har komponenterna några funktioner som underlättar samarbete dem emellan. Är detta tillräckligt för att vi skall kunna tala om ett program istället för en samling komponenter? Hur påverkas situationen av att vi säljer samtliga komponenter i ett paket, men uppmuntrar till användande av enskilda komponenter i utvecklingsarbete?

Här räcker det att peka på problemet och skissa en lösning. Som antydde ovan finns det skäl att anta att funktionaliteten kommer att bli avgörande för dessa frågor. Om en komponent fungerar självständigt, men *dessutom* har en uppsättning specialiserade funktioner, är den nog att betrakta som en enskild komponent. Om en komponent endast kan fungera tillsammans med andra är den att betrakta som en del av en större komponent, kanske av ett program.

3.2 GEMENSAM UPPHOVSRÄTT

En annan fråga som uppstår här är frågan om gemensam upphovsrätt. Om en eller flera personer tillsammans framställer något som är värt upphovsrättsligt skydd så tillkommer skyddet dem alla och de förfogar tillsammans över upphovsrätten.

EXEMPEL 11

Per, Erik och åtta andra programmerare (som nu kommit över sina konflikter och arbetar som egna företagare) samarbetar kring ett projekt om ett stort nytt dataspel som skall bestå av tusen olika rum att gå runt i. Tio oberoende, men samarbetande programmerare med egna företag delar således på dessa tusen rum och konstruerar hundra rum var. Ramverket för rummen och spelet byggs gemensamt. När spelet är färdigt vill Per och Erik sälja det, men de andra iblandade programmerarna invänder att de har upphovsrätt till sina rum och därmed till spelet i sin helhet. Har de rätt?

Med reservation för att det alltid är praktiskt svårt att dra gränser i dessa frågor är svaret att programmerarna har rätt. Enligt 6 § URL gäller som huvudregel att om inte annat är avtalat så äger programmerarna upphovsrätten gemensamt och förfogar över den gemensamt. De måste alltså vara ense om varje för-fogande (försäljning exempelvis). Däremot kan var och en angripa intrång i upphovsrätten i sin helhet.

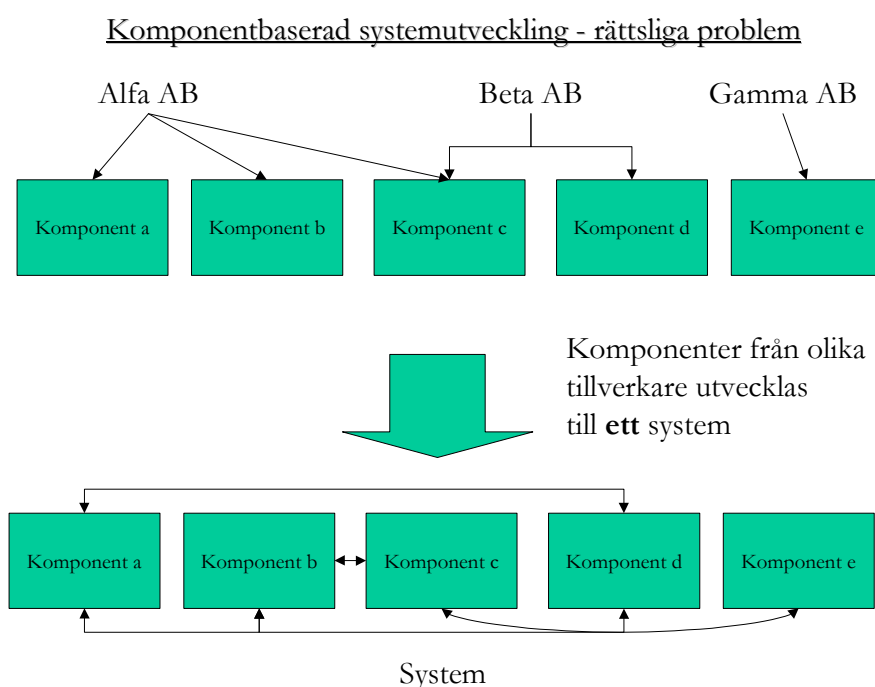
EXEMPEL 12

Per och Erik ”glömmer” spelet på ett kafé där en kollega till dem hittar det och ger ut det. De andra programmerarna blir rasande, men Per och Erik förklarar att de inte vill väcka talan mot sin kollega och att intrånget därför inte kan hindras. Har de rätt?

Nej! Rätten att väcka talan om ansvar för intrång tillkommer var och en enskilt. En upphovsman som äger en del i en gemensam upphovsrätt äger alltså inte hindra de andra om de vill väcka talan om ansvar för intrång.

Vad kunde Per och Erik ha gjort här? Det absolut viktigaste är givetvis att de borde ha avtalat med de andra programmerarna om upphovsrättsliga frågor. Ett väl skrivet avtal kan placera förfoganderätten hos Per och Erik. En annan sak är att om de andra programmerarna endast bidragit med enklare hjälp så erhåller de ej någon upphovsrätt. Den som skriver ut ett manuskript får ingen upphovsrätt. Se också ovan om anställningssituationen.

Detta problem blir större och mer mångfacetterat om vi flyttar in det i en utvecklingssituation med komponentbaserade system. I en bild kan detta åskådliggöras på ett förtjänstfullt sätt.



Vad den här modellen visar är en inte alltför orealistisk situation där ett system utvecklas med hjälp av ett antal komponenter från olika tillverkare. Tre olika företag har utvecklat komponenter som används i systemet. En av komponenterna har utvecklats i samarbete mellan två av företagen.

Vilka problem kan uppkomma här? Först och främst kan det uppkomma problem med upphovsrättigheterna till komponenterna. Samtliga komponenter måste vara klarerade, och användandet av dem måste vara legitimt. Om vi antar att vi inte frågat Gamma AB ovan, eftersom deras produkt spelar en liten roll i systemet, så betyder det att de kan hindra oss att använda deras komponent och därmed systemet. Det spelar alltså ingen roll hur stor del en komponent spelar i ett system – komponenten måste fortfarande användas i enlighet med upphovsrättsliga regler.

Problemen som uppstår vid komponentbaserad utveckling påminner om dem som vi finner vid framställandet av multimedieprodukter där ett stort antal enskilda verk kan ingå (se Westman, Lindberg s 117).

Ett annat problem uppstår om inte Alfa AB och Beta AB har avtalat om komponent c, den komponent som de utvecklat gemensamt. Då måste båda företagen gå med på att sälja produkten. Om vi som systemutvecklare endast avtalat med den ena parten, kan den andra parten dra tillbaka produkten och väcka talan mot oss. Vän av ordning undrar kanske varför de skulle göra det? Antag att Alfa AB utvecklat en separat produkt, komponent f, som de anser vara bättre! Då är det naturligt att de vill stoppa Beta ABs försäljning av komponent c.

Vad blir konsekvensen av att en enda länka i kedjan någonstans spricker? Systemet i sin helhet blir då obrukbart. En följd som kan få drastiska effekter och kosta utvecklarna och beställarna stora summor.

Hur kan vi då lösa dessa problem, som samtliga har den allvarliga konsekvensen att systemet inte kan utnyttjas? En enda enkel regel är här på sin plats:

Avtala alltid om utnyttjande av komponenter med alla relevanta parter!

Så länge vi kommer ihåg detta, att utreda vilka som äger en komponent och avtala med denne om användandet av komponenten, så är vi på den säkra sidan.

Därefter kommer ytterligare ett problem. Systemet i sin helhet kan uppvisa en betydande originalitet.

EXEMPEL 13

I en bok presenterar den engelske författaren Douglas Adams en produkt som gör musik av bokföringssiffror och årsresultat. Låt oss tänka oss att ett sådant system kan tillverkas ganska lätt genom att vi sammanfogar en eller flera komponenter för bokföring med en musikkomponent. Vilket skydd har detta system?

Det har här uppkommit ett nytt verk som är tämligen originellt. Det har alltså ett eget upphovsrättsligt skydd. Systemutvecklaren kan förfoga över systemet, sälja det eller licensiera ut det, men hans rätt är beroende av de rättigheter som rättighetsinnehavarna bakom komponenterna har! Om han inte har avtalat med dem så kan han inte utnyttja sin upphovsrätt alls! Det klassiska exemplet är kompositören som tonsätter en diktares hela verk, bara för att få veta att diktares inte vill bli tonsatt. Sångerna eller musiken kan då aldrig framföras.

Inte nog med att det finns en massa ursprungliga rättigheter att ta hänsyn till, alltså. Vi måste också beakta de rättigheter som uppstår i sammansättandet av komponenterna. Och här kan vi konstatera att det inte räcker att sätta samman en Ikeaprodukt för att erhålla skydd. Det krävs ett visst mått av originalitet.

3.3 ÖVRIGA FRÅGOR

Det finns ett stort antal frågor som cirkulerar i bakgrunden, när vi diskuterar komponenter. Det vore inte korrekt att bortse från dessa. Däremot skulle det föra för långt att diskutera dem för djupt. Några översiktliga drag kan dock vara intressanta att granska.

3.3.1 Komponenter och skadestånd

Antag att vi har avtalat om ett antal komponenter med en mängd olika parter och att en av dessa ljög om sin rätt till en komponent. Den riktige rättighetsinnehavaren kontakter oss och begär att vi slutar använda hans produkt. Vad kan vi då göra?

Svaret på denna fråga beror delvis av vilket avtal vi har med den som sålt komponenten. Om vi i detta avtal skrivit in en skadeståndsklausul uppstår i stort inga problem alls. Det följer förmodligen av den klausulen av leverantören är ansvarig för den här typen av ”fel”.

När det gäller summan för skadeståndet kan den också avtalas om i viss utsträckning. Förmodligen kommer domstolen också att ta hänsyn till vad som kallas det positiva kontraktsintresset. Detta innebär att den som förlorar på att en annan part förfarit svikligt skall få sin ekonomiska position helt återställd. Skadeståndet beräknas på så sätt att den som ”blivit lurad” får samma ekonomiska ersättning som försäljningen av systemet skulle kunnat ge. Detta är givetvis svårt att beräkna. Domstolen kan också ta hänsyn till extra licenskostnader för att skaffa en ersättnings komponent, m.m.

3.3.2 Komponenter och agenter

En annan intressant fråga är vad som händer om vi kombinerar komponenter med mer eller mindre intelligenta agenter. Hittills har vi förutsatt att komponenterna skulle vara små byggstenar för att skapa program och att de knappast själva skulle vidta åtgärder. Om vi istället gör oss bilden av ett flertal specialdesignade myror som kan kommunicera och arbeta tillsammans får vi en helt annan typ av problematik.

Intelligenta agenter är ett aktuellt forskningsområde och mycket händer inom det. I en möjlig framtid kan vi skapa komponenter som interagerar med de andra komponenter som för ögonblicket behövs. I en nätverksmiljö kan min kalender söka upp din och se om vi kan stämma ett möte, mitt textredigerare kan gå ut på det globala nätet och leta reda på en komponent som översätter svenska till kinesiska, eller mer troligt: en komponent som översätter svenska till engelska och ytterligare en som översätter från engelska till kinesiska. Denna myriad av komponenter kommer då att efter behov söka upp andra och under ett fåtal sekunder skapa konstellationer eller system som behövs. Förebilden hittar vi i hjärnans synapser som kopplar och sammanbinder information associativt på olika sätt. Vilka konsekvenser kommer detta att få för juridiken?

Svaret är givetvis enkelt: det kan vi inte veta. Den typ av verklighet som jag beskrivit här aktualiserar en mängd frågor om klassiska upphovsrättsliga begrepp. Vem är skapare av ett system som uppstått på detta sätt? Är det jag? Eller rentav datorn? Idag brukar det regelmässigt stå i läroböcker att en upphovsman måste vara en fysisk, *naturlig*, person. För tio år sedan stod det endast att upphovsmannen måste vara en fysisk, till skillnad från en juridisk, person. Idag är *artificiella* upphovsmän ingen omöjlighet.

Det finns program som med lätthet kan simulera konsertverk i *Mozarts stil*. Vem är upphovsman till dessa verk? Datorn? Programmeraren? Mozart? Och om en programmerare släpper ut ett antal komponenter på det globala nätet som bygger något exceptionellt som han inte kunde förutse – är då denne programmerare verkligen upphovsmannen?

Framtidens situation påminner mer om odlarens och trädgårdsmästarens än om författarens. Den totala kontroll över det egna verket som författaren har kommer att tillhöra det förflutna. Programmeraren kommer att så frön i de globala nätverken. Hur han sedan skördar sina frukter är osäkert i dagsläget.

3.3.3 Komponenter och bevisning

En alltid lika central fråga är hur vi till *syvende og sidst* kan bevisa vad vi påstår. Ofta bortses från denna centrala fråga, men det är trots allt så att oavsett hur rätt vi har så gagnar det oss föga om vi inte kan styrka det inför domstol.

EXEMPEL 14

Per presenterar sitt nya spel på en mäsas i Göteborg. Samtidigt presenterar Erik samma spel i Stockholm. Båda påstår att de skrivit det och att den andre bara ljuger. Hur kan vi lösa denna konflikt?

Här finns inga enkla tumregler och inga självklara svar. Sannolikheten för att båda åstadkommit exakt samma spel är tämligen liten. Frågan är hur vi avgör vem som ljuger?

Ofta kommer det att handla om ett ingående forskningsarbete och om bevisning. Per kanske kan presentera vittnen från spelbolaget som han tänkt sälja till att han demonstrerat spelet tidigare under året, och Erik kanske inte kan förklara hur någon del av spelet är uppbyggd. Domstolen får väga bevisningen som de båda parterna presenterar och sedan försöka utröna vad som stämmer och vad som inte stämmer.

Generellt sett kan det vara bra att sprida, inom en begränsad krets, vad vi håller på med för att på det sättet kunna säkra bevisning om att vi gjort det. Paradoxalt nog blir vi sårbarare för den redovisade typen av bedrägeri om vi håller ett projekt helt hemligt. Vem skall då kunna vittna för att vi faktiskt arbetat med det under de månader som behövts?

Ett sätt att säkra bevisning skulle kunna vara att deponera koden hos en tredje part, under arbetet med den. Detta hjälper emellertid inte mycket. Som tidigare nämnts i bl a *Informationsambällets juridik* (SISU-publikation 98:03) krävs inget särskilt registreringsförfarande för att få upphovsrätt till ett datorprogram. Det är därför tveksamt om domstolen skulle godkänna enbart deposition som ett säkert bevis på upphovsmannaskap. Depositionen kan istället tjäna som ett bevis som i likhet med alla andra måste vägas och analyseras.

3.3.4 Komponenter och global handel

Ett mål med den utveckling som lett till komponentbaserad systemutveckling, är att försöka minska behovet av att uppfinna hjulet om och om igen. Ofta nog har något redan gjorts och då kan det vara ekonomiskt riktigt att köpa en komponent istället för att skriva om samma komponent själv.

Givetvis vill vi då kunna utnyttja hela världens programmerare! De programmerare som finns i Sverige är säkerligen duktiga, men att vara begränsad till dem och deras produkter i sökandet efter lösningar är inte önskvärt.

Komponenthandeln bör kunna ske globalt. Frågan uppkommer då om andra rättssystem ser ut som vårt? Hur är det med rättigheterna till program runtom i världen?

Det skulle här föra för långt att redogöra i detalj för varje större rättssystem. Vi kan dock konstatera att inom den Europeiska Unionen är integrationen tämligen god (se Rosén och Bainbridge). De regler som vi sett här gäller i större eller mindre utsträckning i EU. Vi räknar ofta med två större rättssystem i världen idag. Kontinental och anglosaxisk rätt. Sverige brukar räknas till den kontinentala delen (tillsammans med bl a Tyskland). Till den anglosaxiska räknas Storbritannien och USA främst. Synen på immaterialrätt i dessa rättssystem skiljer sig litet åt.

Ansträngningar görs dock på området, främst genom olika internationella organisationer som WIPO och WTO. (Se <http://www.wipo.org> och <http://www.wto.org>).

Den som önskar sälja sina komponenter på en global marknad måste göra mycket noggranna förberedelser med avtal och kanske en lättare undersökning av de marknader som han eller hon riktar sig mot. Det är ännu betydande väg kvar till en världsgemensam immaterialrätt.

3.3.5 Reverse engineering och dekompilering

I korthet skall också nämnas de frågor som behandlar s k *reverse engineering*. Vid *reverse engineering* tar någon isär ett program för att förstå hur det fungerar och skriver sedan om det i en annan form.

26 § g URL stadgar att det är tillåtet att göra detta. Denna rätt kan inte heller avtalas bort (Koktvedgaard, Levin s. 156 ff).

En annan fråga är om vi får dekompilera ett program alls (att dekompilera ett program är att föra det från objektкод till källkod, alltså från maskinläsbar form till människoläsbar form). Denna fråga aktualiseras kanske oftare med komponenter än med andra program. I 26§ h URL regleras dekompilering. I korthet sägs att dekompilering får ske om det sker för att möjliggöra programmets interoperabilitet. Snävheten i detta stadgande är viktig.

Dessa frågor prövas sällan, men kan kanske komma att bli mer betydelsefulla i samband med att komponentmarknaden växer.

4 EN MARKNAD FÖR KOMPONENTER?

Under arbetet med denna rapport föddes en mängd tankar kring hur framtidens handel med komponenter kan se ut. Detta kapitel är resultatet av detta skissande.

Vad som framförallt är intressant är att se vilka strukturer som finns idag på andra områden där immaterialrättigheter licensieras och säljes.

Modellerna som redovisas här kan vara intressant att se på ur affärsutvecklingsperspektiv. Många företag som kommer att ägna sig åt informationsmäklari i framtiden kanske också kan lägga till komponentmäklari till sitt tjänsteutbud.

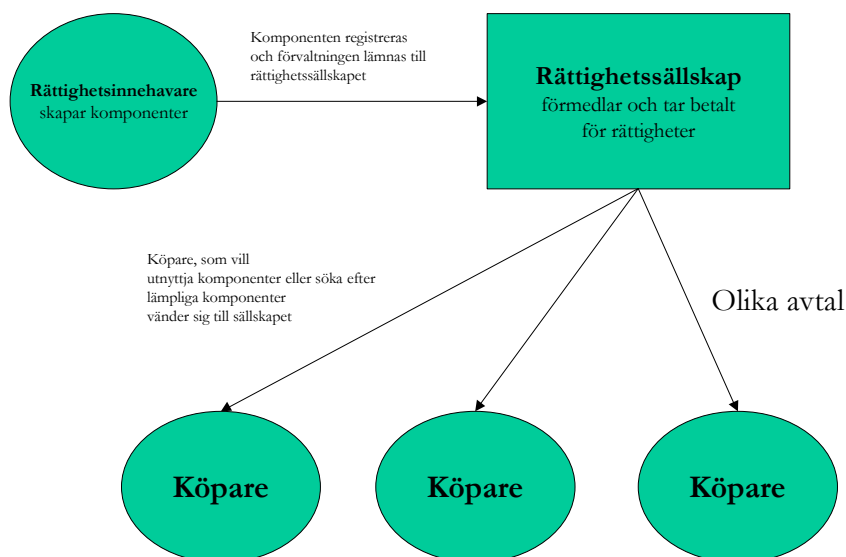
En annan sak modellerna försöker visa är effektiva sätt att handla med komponenter.

4.1 MODELLER

Det område som ligger närmast till hands att studera idag om vi vill hitta en modell för hur vi kommer att handla med komponenter i framtiden är handel med till exempel musikrättigheter och bildrättigheter. Denna handel bedrivs sedan ett tag tillbaka på ett mycket enkelt sätt via olika rättighetssällskap som förmedlar och tar betalt för att förmedla rättigheter.

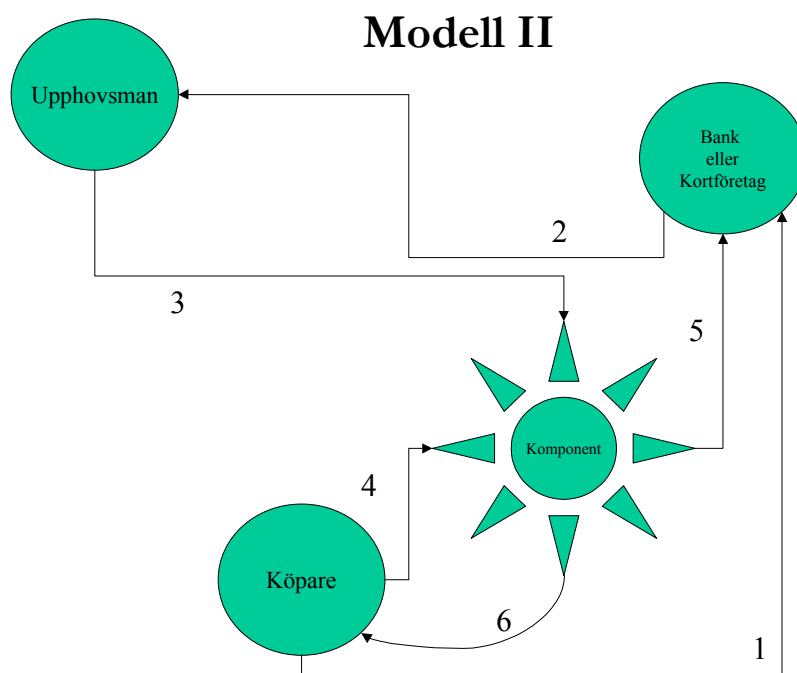
Detta tillvägagångssätt går givetvis att tillämpa även avseende komponenter. I en bild:

Modell I



Denna centraliserade marknad liknar mycket den marknad som finns idag för musik- och bildrättigheter. Den har både för- och nackdelar. Den tydligaste fördelen med denna modell är att den som skrivit en komponent kan kontrollera bruket av komponenten ganska bra. Nackdelen är att det krävs väldiga databaser och ett stort mått av kontrollverksamhet för att rättighetssällskapet skall kunna fungera. Att registrera alla avtal och köpare tar förstås en väldig tid. De administrativa kostnaderna riskerar därmed att bli mycket höga.

Vi kan också tänka oss en modell där komponenterna själva läser upp sig efter att ha fått en betalningsbekräftelse. Denna modell är mer futuristisk, men inte därför mindre intressant:



I den här modellen är det komponenten själv som fyller rättighetssällskapets roll. Vi kan gå steg för steg genom modellen. Efter att ha laddat hem komponenten och utvärderat den bestämmer sig köparen för att använda den. Han kontakter då en bank eller ett kortföretag och gör en betalning (steg 1). Denna betalning vidarebefordras sedan till upphovsmannen (steg 2) som, kanske automatiskt, skickar en kod till komponenten som talar om att köparen har rätt att använda den (steg 3). Köparen begär sedan att komponenten läser upp sig (steg 4). Komponentens kontrollerar köparens identitet med kortnumret och att betalning skett en extra gång (steg 5) och läser sedan upp sig (steg 6).

Allt detta kan givetvis byggas på så sätt att det blir närmast omärkligt för köparen.

Båda dessa modeller ger upphov till en hel del avancerade rättsliga frågor. I modell I är *en* enkel fråga vad som händer om en köpare vill sälja en komponent till en annan köpare utan att gå via rättighetssällskapet. Avtalet blir bestämmande för om detta är tillåtet eller ej. I modell II uppkommer till exempel ansvarsfrågor. Vad händer om banken inte förmedlar rätt information och komponenten inte läses upp? Även detta kan regleras i avtal.

Det låter sig inte sägas vilken av modellerna som är bäst, men det låter sig sägas att den första modellen har större rättslig förankring, eftersom den tar sin förebild från en redan existerande handel med immaterialrättigheter.

Det finns anledning att studera de avtal och de former för handel med komponenter som uppstår. Dels är det viktigt att denna handel växer fram med en förståelse för den juridiska traditionen, men det är lika viktigt att den juridiska traditionen inte försvårar nya handelsformer. Kan utvecklingen ske symmetriskt är det givetvis önskvärt.

5 SLUTSATSER

Det kan var intressant att som avslutning göra några korta kommentarer om vad detta studium leder oss till för slutsatser. En viktig slutsats är att det ständigt utvecklas nya former av teknik som inte förutsetts eller reglerats av lagstiftaren. Vad vi tvingas till i de situationer där detta händer är att försöka undersöka vilken lagstiftning som ligger närmast och hur den kan användas.

I fallet med komponentbaserad systemutveckling är situationen förhållandevis god. Vi har ett existerande, tämligen internationellt, ramverk för datorprogram som vi kan lägga som en grund för de avtal och andra regleringar som vi önskar på området.

När det gäller andra områden – som den framväxande elektroniska handeln – så är det sämre ställt. Våra redovisningslagar kräver fortfarande *papper* i stor utsträckning och konsumenternas ställning i den globala handeln är ett endast bitvis utforskat område.

När våra lagar inte kan hantera en ny företeelse utlämnas vi åt avtal och avtalsskrivningens ofta svåra konst. Ofta nog är det avskräckande komplicerat att formulera ett avtal som känns riktigt för båda parter. Men konsekvensen av att inte ha formulerat det när tvist uppkommer är att vi utsätter oss för den osäkerhet som uppstår när en gammal lagstiftning möter ny teknik.

Lösningen på detta problem är inte heller att stifta nya lagar. Det finns en betydande tröghet i det rättsliga systemet som är ägnad att skydda oss. I en stat där lagarna kan ändras över natten vilar ingen tryggt. Att ställa kravet på lagstiftaren att han tillägnar sig all den nya tekniken och lagstiftar om den är vare sig önskvärt eller möjligt.

Istället måste vi hitta nya lösningar. Branschavtal, självreglering, etiska rekommendationer Alla dessa former kan ändras snabbt och följa med utvecklingen. Dessutom har de fördelen av att vara formulerade av yrkeskunniga människor som förstår sitt ämne.

Den komponentbaserade systemutvecklingen kan rättsligen förutspås en lugn och god framtid om branschorganisationer och andra intresserade tar hänsyn till de nya problem som uppstår och reglerar dem i olika branschavtal och i god affärssed. Härvid kan mycket vinnas om en utblick mot multimediaområdet görs, eftersom problemen på det området är snarlika.

6 LITTERATUR

Det finns mycket litteratur att tillgå på detta område och de böcker jag här rekommenderar är sådan som jag själv har haft nytta av i arbetet med denna rapport.

Bainbridge, David *Software Copyright Law* 3:rd ed London 1997

Koktvedgaard, Mogens och Levin, Marianne *Lärobok i immaterialrätt* 5 uppl Stockholm 1997

Olsson, Henry *Copyright: svensk och internationell upphovsrätt* 6 uppl. Stockholm 1998

Rosén, Jan *Swedish Software Law: As Related Primarily to EC Directives*

Westman, Daniel och Lindberg, Agne *Praktisk IT-rätt* Stockholm 1997