

Borde den svarta lådan vara grå?

Grey box-principen minskar missförstånden



2012-06-28: Sven-Håkan Olsson

VAD TILLFÖR GREY-BOX? Tanken med black box är bra, men inte sällan kan man komma runt missförstånd genom att ökad transparens – inte minst i stora organisationer.

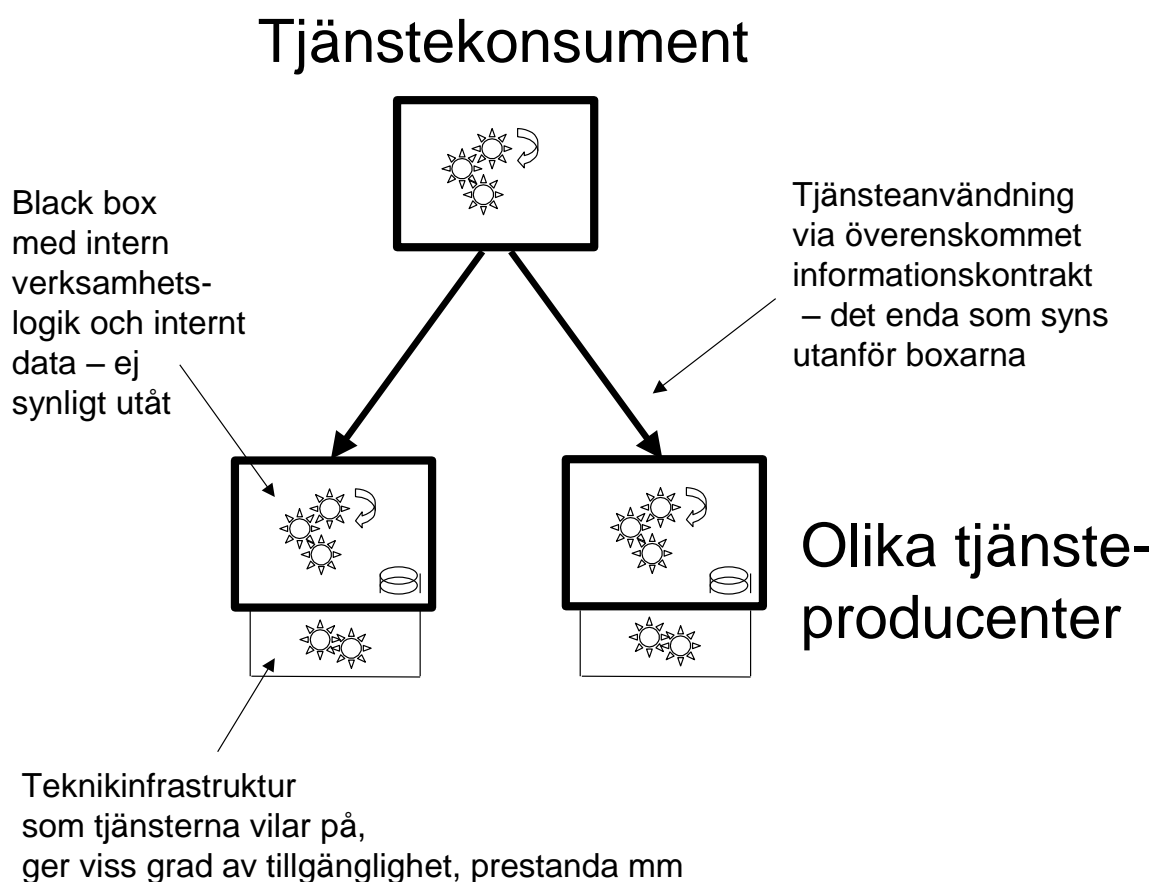
Begreppet ”black box” har länge gjort nytta för att göra ansvarsgränser tydliga och minska mängden beroenden. Analogin står för att innanmätet i en låda inte ska vara synligt utåt, det är endast några externt dokumenterade egenskaper som omvärlden ska behöva få reda på. Den som skapar innanmätet för därmed en mycket större frihet och slipper myriader av krångliga detaljberoenden gentemot omvärlden.

Uttrycket black box har använts i många helt olika sammanhang, som till exempel inom bilelektroniken där en liten svart låda kan sköta till exempel den elektroniska tändningen medan bilmekanikern inte behöver veta något om lådans innanmäte. Inom programmering blev det mer känt i samband med populariseringen av objektorientering i början på 90-talet, då ofta med synonymen inkapsling. Black box-uttrycket används även i sammanhang som it-drift (kunden behöver inte veta hur leverantören utformat driftmiljön, bara vad som avtalats i form av tillförlitlighet, prestanda osv) och testning (testaren ska inte veta om hur innanmätet är utformat, bara att det resulterar i vad som överenskommits).

Krävs mer än en syntaxbeskrivning

När Service Oriented Architecture (SOA) var som hetast för ett antal år sedan betonades ordet kontrakt för lådans överenskommelse med omvärlden vilket var bra men man tyckte ofta att det räckte med en WSDL-fil och den beskriver bara syntaxen för användning av en tjänst. Endast en syntaxbeskrivning räcker inte alls, man måste också komma överens om semantik och begreppsdefinitioner, upptid/prestanda/tillförlitlighet, katastrofskydd, kostnadssättning för tjänsten, juridiska förutsättningar, processförhållanden, långsiktig vidareutveckling av tjänsten med mera. Då blir den svarta lådans verkan och funktion väl definierad. (I denna

artikel behandlas för övrigt endast tjänster som maskiner använder, inte tjänster direkt för personer.)



Rätt tänkt men det finns nackdelar

Black box-tänkandet har varit till stor fördel. Men det finns några nackdelar, främst:

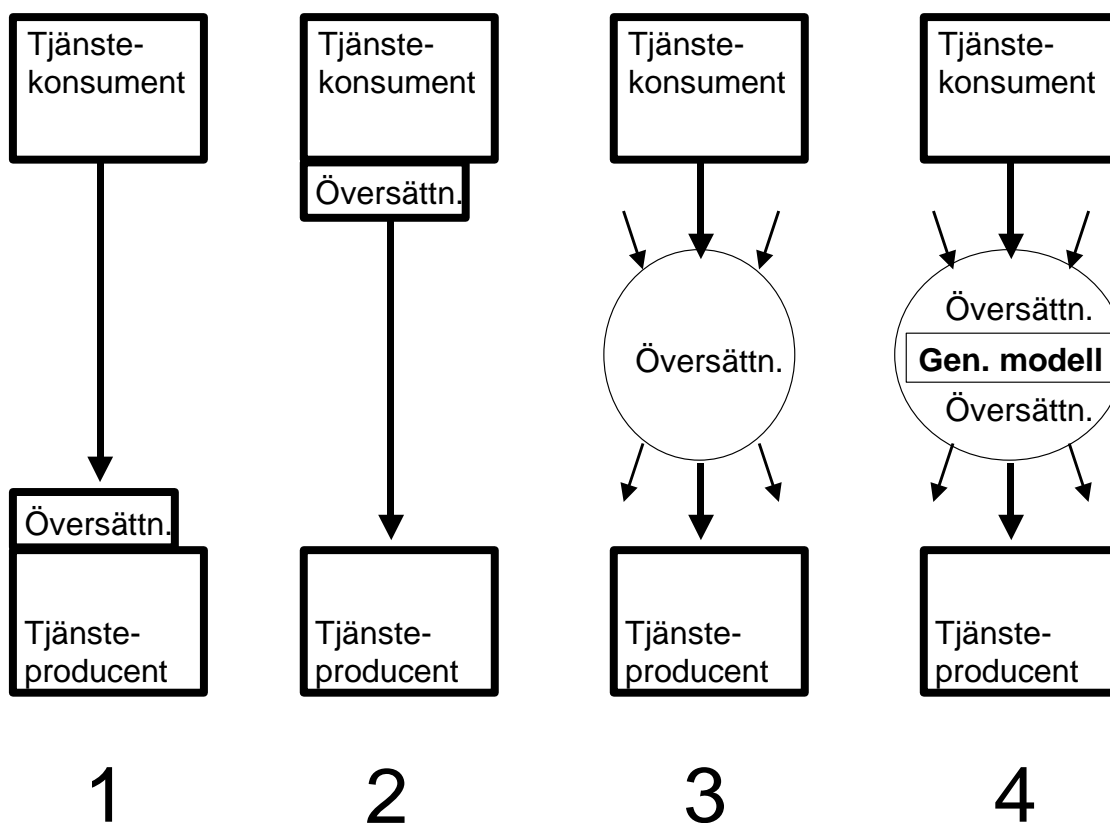
- Missförstånd mellan verksamhetskunniga inom olika black boxes som ska samverka
- Övertro på kontraktets värde juridiskt/affärsmässigt (detta planerar jag att behandla i en kommande trendspaning)

Den första nackdelen handlar om att en större black box ofta motsvaras av en viss verksamhet, en avdelning eller ett dotterbolag som är specialiserat på en viss sak. Den sortens stora svarta lådor kallar jag ofta för SOA-domäner. Ta till exempel kreditkortsavdelningen inom en bank. Verksamhetskunnandet där är mycket specifikt och man behöver egna begrepp som inte nödvändigtvis finns inom andra delar av banken, såsom inom inlåningsavdelningen.

Ändå behöver man ofta integrera information och funktion mellan separata svarta lådor. Det kan vara för att betjäna kunden mer sammanhållet i en mobil-app, det kan vara för att ett kundcenter ska ges bättre chans till merförsäljning, det kan vara för att få sammanhållen statistik för hela bankkoncernen, för att nämna några anledningar.

När integration ska införas är det mycket vanligt att syntax/begrepp/informationsmodell inte riktigt stämmer mellan de svarta lådorna. Alltså måste översättning ske. Här finns det några olika principer (andra kombinationer är också tänkbara):

1. Aktuell **tjänsteproducent tar ett extraansvar** och översätter till tjänstekonsumentens önskade syntax/begrepp. Tjänsteproducentens svarta låda har därvid utökats något.
2. Aktuell **tjänstekonsument tar ett extraansvar** och översätter till tjänsteproducentens levererade syntax/begrepp. Tjänstekonsumentens svarta låda har därvid utökats något.
3. En **mellanhand introduceras** som gör översättningen mellan levererad och önskad syntax/begrepp. Detta har varit populärt ett antal år och kallas ofta nav eller buss, EAI (Enterprise Application Integration) eller ESB (Enterprise Service Bus).
4. Som i föregående princip så skapas en mellanhand men på ett ännu mer ambitiöst sätt, den översätter två gånger, **via en generell informationsmodell** ”på vägen”. En sådan generell modell är tänkt att vara det gemensamma språket, koncernens esperanto. Ett namn som ofta används är kanonisk modell. Denna variant har också varit populär ett antal år.



Brist på insyn komplicerar kommunikationen

Vad var det nu för nackdel jag ville ta upp? Jo, missförstånd, framförallt i de fall då ett informationskontrakt inte direkt motsvaras av en viss svart lådas interna informationsmodell och logik, det vill säga vid majoriteten av översättningsprinciperna ovan.

När koncernens affärsutvecklare funderar över nya kundkanaler, övergripande prissättningsidéer eller vad det nu kan vara, vill de tala med verksamhetskunniga inom inlåning och inom kreditkort, spåna fram vad som är möjligt och vettigt utifrån deras olika och specifika verksamhetskunnande. Men om black box-tänkandet är strikt så får de verksamhetskunniga inte prata om sitt innanmäte med parter utanför en sin egen svarta låda!

Missförstånden riskerar då bli många eftersom de verksamhetskunniga kan sina interna begrepp, men är ofta mindre kunniga i hur deras begrepp har översatts till en generell informationsmodell eller till andra svarta lådors modeller. Integratören däremot har informationskontrakten för black box att arbeta med och de är ofta annorlunda, vilket leder till missförstånd eller tidsödande utredningar. Viss precisionsförlust kan också uppstå vid översättning till en generell modell.

En förbättring vid de här problemen är vad som har kommit att kallas **grey box**. Tanken är att lådas hölje inte ska vara helt svart utan man ska kunna få se in i viss mån. Men, vänta nu, ett grått hölje är väl inte nödvändigtvis halvgenomskinligt? Ordet ger alltså egentligen fel analogi, men nu har det slagit igenom, så det är väl bara att acceptera.

Enligt grey box-principen är det alltså tillåtet för externa parter att få reda på en del om innanmätet i en låda. Kanske publicerar man de olika mappningarna mellan extern och intern informationsmodell. Grey box-tänkandet gör det lättare för verksamhetskunniga och affärsutvecklare att förstå vilka outnyttjade potentialer som kan finnas, och det gör det lättare att sedan konstruera korrekt översättningslogik. Även testning och felsökning blir enklare.

Fallet jag har exemplifierat med ovan är en större organisation som har olika SOA-domäner internt. Om man däremot kopplar ihop sig med tjänster i helt andra organisationer är grey boxing mer tveksam. Då får man nog acceptera att förhandla kring ”officiella tjänstegränssnitt” och använda en striktare black box-princip.

En helt annan viktig sak är att definiera vem som ansvarar för översättningsarbetet i nav eller buss från/till en viss black box-modell. Ska kreditkortsavdelningen bära ansvaret, trots att det sker i en buss som kreditkortspersonalen inte har kunskap om eller teknisk kontroll över? Eller är det någon viss avdelning som hanterar bussen, kanske ett ICC (Integration Competence Center) som ska ansvara? Blir bussen föresten en egen black box på det här viset? Eller blir det en grey box? Man kanske skiljer på ägaransvar och utföraransvar för översättningsarbetet?

Lästips:

- Black box-principen tas upp i min tidigare trendspaning [Hur den lösa kopplingen ändå blir hård](#) men ur en lite annan aspekt.



Sven-Håkan Olsson sysslar just nu med en iPad-tjänst för bolagsstyrelser och nämnder. I övrigt är han oberoende konsult som särskilt arbetar med att kombinera verksamhetsnytta med teknikhöjd. Han har en lång karriär sedan 70-talet som it-konsult (applikationsarkitektur, systemdesign, programmering, reviewer, utredningar, kursledning). Sven-Håkan är också medgrundare av Know IT och var dess teknikchef 1990-2003. Han utsågs till en av "Sveriges topputvecklare" av Computer Sweden 2008 och 2010.

Sven-Håkan håller regelbundet kurser åt Dataföreningen. Läs gärna mer på hans blogg www.definitivus.se samt på www.styrelsemote.se .

[Sven-Håkan Olsson](#)