

COMPOS – ett Programmeringsspråk för att Koppla Ihop Smarta Prylar

Datorer som kan kopplas upp mot internet blir allt billigare och mindre. Detta gör att de kan integreras i produkter för att skapa så kallade smarta prylar. Exempel på prylar som kan vara smarta är lampor, termostater, vågar, blodtrycksmätare, lås etc. Denna trend med uppkopplade smarta prylar kallas sakernas internet. Sakernas internet kan utnyttjas på olika områden, så som i hemmet, sjukvården, industrin eller inom jordbruket. Genom att koppla ihop flera smarta prylar skapas ett så kallat sakernas-internetsystem. I dessa system samarbetar olika prylar för att bli till större nytta än varje pryl för sig själv. Ett exempel på ett sakernas-internetsystem är ett smart lås kopplat till smarta lampor, så att när du låser upp ditt hem, tänds dina lampor.

I denna avhandling vill vi göra det enklare att utveckla sakernas-internetsystem. I de system vi undersöker kan prylarna berätta vilka meddelande de kan skicka och ta emot. Sedan finns det små datorprogram, kallade kompositioner, som kopplar ihop prylarna. För att specificera kompositioner, har vi i avhandlingen skapat och utvärderat ett programmeringsspråk med detta som enda syfte. Vårt programmeringsspråk heter COMPOS. Genom att göra ett programmeringsspråk med ett enda syfte kan vi bortse från många funktioner som vanligen finns i mer generella programmeringsspråk och på så sätt göra programmen enklare att förstå, använda och analysera.

Koden nedan visar en komposition skriven i COMPOS som tänder två lampor om någon låser upp ytterdörren. Kompositionen kan köras på vilken dator som helst, t.ex. det smarta låset eller Wi-Fi routern. Den första raden instruerar datorn att vänta på att låset ska meddela att någon har låst upp dörren. Rad 2-3 instruerar datorn att skicka meddelande för att tända lampan i hallen och i köket. Att rad 2-3 är indragna betyder att de utförs först efter det på rad 1 har hänt.

```
1 when receive låst upp from ytterdörrens lås do
2     send tänd to lampa hallen
3     send tänd to lampa köket
```

En skillnad mellan ett system på en enda dator och ett sakernas-internetsystem, är att de senare måste hantera att prylar tappar kontakten och kanske senare får kontakt igen. Vi har designat COMPOS så att när prylar tappar kontakten, blir det en begränsad påverkan på resten av systemet. T.ex. om lampan i hallen tappar kontakten med systemet, ska lampan i köket fortfarande tändas när man låser upp sitt smarta lås.

Sakernas-internetsystem blir lätt komplexa med prylar som skickar meddelanden kors och tvärs. Genom att ha kompositioner som beskriver hur meddelandena flödar i systemet kan vi analysera dem för att få inblick i hur ett system fungerar. I denna avhandling har vi visat hur körande system kan modelleras och analyseras på en hög nivå. Som exempel har vi skapat en analys som räknar ut

vilka delar av systemet som fungerar när olika prylar tappar kontakten. Detta kan till exempel vara användbart om man vill försäkra sig om att lampan i köket tänds, även när lampan i hallen har tappat kontakten.

För att vidare underlätta utvecklingen av sakernas-internetsystem, har vi integrerat COMPOS i en programmeringsmiljö som tillåter användaren att se vilka prylar som är uppkopplade. Användaren kan sedan "dra" en uppkopplad pryl till COMPOS-editorn för att använda prylen i en COMPOS-komposition. Som en del i att utveckla programmeringsmiljön, har vi skapat och utvärderat ett editorramverk, som heter JATTE. JATTE kan användas till olika programmeringsspråk och integreras i olika programmeringsmiljöer.

För att utvärdera COMPOS, har vi tagit ett sakernas-internetsystem för hem-sjukvård av njursviktpatienter, skrivit med ett annat kompositionsspråk, och åter-skapat kompositionerna med COMPOS. Jämfört med ursprungliga kompositionsspråk är meddelandeflödet i COMPOS tydligare. Fördelarna med COMPOS är att man enklare kan programmera komplicerade meddelandeflöden och skapa analyser. Med denna forskning hoppas vi kunna bidra till att göra det enklare och och snabbare att utveckla sakernas-internetsystem.