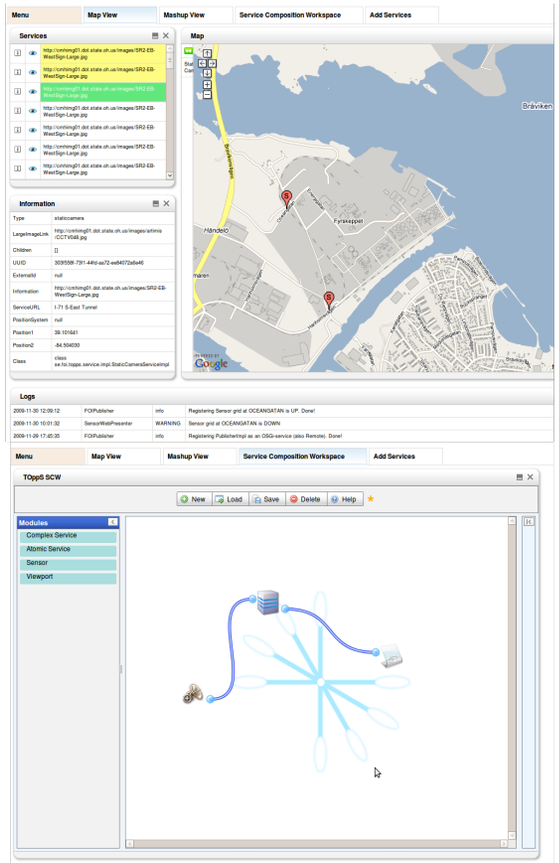
Dynamiskt Tjänstebaserat Ramverk för Opportunistiska Sensorer och Sensornätverk (TOppS), Slutrapport

MARIANELA GARCÍA LOZANO, PONTUS HÖRLING, FARSHAD MORADI,
PETER SIGRAY, EDWARD TJÖRNHAMMAR



FOI är en huvudsakligen uppdragsfinansierad myndighet under Försvarsdepartementet. Kärnverksamheten är forskning, metod- och teknikutveckling till nytta för försvar och säkerhet. Organisationen har cirka 1000 anställda varav ungefär 800 är forskare. Detta gör organisationen till Sveriges största forskningsinstitut. FOI ger kunderna tillgång till ledande expertis inom ett stort antal tillämpningsområden såsom säkerhetspolitiska studier och analyser inom försvar och säkerhet, bedömning av olika typer av hot, system för ledning och hantering av kriser, skydd mot och hantering av farliga ämnen, IT-säkerhet och nya sensorers möjligheter.

FOI
Totalförsvarets forskningsinstitut
Informationssystem
Box 1165
581 11 Linköping

Tel: 013-37 80 00
Fax: 013-37 81 00

www.foi.se

FOI-R--2860--SE
ISSN 1650-1942

Användarrapport
December 2009

Informationssystem

Dynamiskt Tjänstebaserat Ramverk för Opportunistiska Sensorer och Sensornätverk (TOppS), Slutrapport

Framsidesbilden visar en kombinerad vy av Webbgränssnittet. Det som kan ses i övre delen av figuren är kartvyn där tillgängliga tjänster samt information om dem ses. I den nedre delen syns tjänstesammansättningsvyn där tjänster kopplas ihop och nya tjänster byggas.

FOI-R--2860--SE

Marianela García Lozano, Pontus Hörling, Farshad Moradi, Peter Sigray, Edward Tjörnhammar

Dynamiskt Tjänstebaserat Ramverk för Opportunistiska Sensorer och Sensornätverk (TOppS), Slutrapport

Titel	Dynamiskt Tjänstebaserat Ramverk för Opportunistiska Sensorer och Sensornätverk (TOppS), Slutrapport
Title	Dynamic Service-oriented Framework for Opportunistic Sensors and Sensor Networks (TOppS), Final Report
Rapportnummer / **Report no**	FOI-R--2860--SE
Rapporttyp / **Report type**	Användarrapport / User report
Utgivningsår / **Year**	2009
Antal sidor / **Pages**	48
Kund / **Customer**	FM
Forskningsområde	2. Operationsanalys, modellering och simulering
Research area	2. Operational Research, Modelling and Simulation
Delområde	21. Modellering och simulering
Sub area code	21. Modelling and Simulation
Projektnummer / **Project no**	E7122
Godkänd av / **Approved by**	Martin Rantzer Chef, Informationssystem
ISSN	ISSN-1650-1942

FOI Totalförsvarets forskningsinstitut
Informationssystem
Box 1165
581 11 LINKÖPING

Sammanfattning

Detta är slutrapporten för projektet TOppS - Dynamiskt Tjänstebaserat Ramverk för Opportunistiska Sensorer och Sensornätverk. Projektet har genomförts som en strategisk forskningskärna på FOI under åren 2007 - 2009. TOppS uppgift har varit att ta fram en arkitektur och en prototyp till ett tjänstebaserat ramverk där heterogena sensorer kan ansluta sig och göras åtkomliga som informationstjänster för operatörer i ledningssystem.
TOppS vision, liksom liknande initiativ i omvärlden, beskrivs. En arkitektur för att möjliggöra denna idé har tagits fram liksom en prototypimplementation vilket beskrivs i eget kapitel. Samarbeten som projektet har varit inblandat i beskrivs även, liksom två publikationer vilka producerats inom TOppS projektet. Projektet har även finansierat fyra examensarbeten inom området, vilka omnämns kort. Sammanfattningsvis avslutas rapporten med ett kapitel innehållandes en diskussion kring utmaningar och vinster med TOppS samt de erhållna resultaten.

Nyckelord

SOA, Opportunistisk, Sensor plug-and-play, Sensorsjälvregistrering, Informationstjänster, Sensorplanering

Abstract

This is the final report in project TOppS - Dynamic Service-oriented Framework for Opportunistic Sensors and Sensor Networks. The project has been conducted as a so-called strategic research core at FOI during 2007 - 2009. TOppS objective has been to present an architecture and a prototype for a service-oriented framework where heterogeneous sensors can connect, register, and be made reachable as information services for operators in Command and Control (C2) systems.
The vision of TOppS, as well as similar initiatives around the world, are described. An architecture to enable the TOppS paradigm is proposed together with a prototype implementation which is described in its own chapter. Cooperations that the project has been involved in are described, as well as two publications that have been produced within the TOppS project. The project has financed four thesis works which are briefly described. In conclusion the report is summarized with a chapter discussing challanges, benefits and project results.

Keywords

SOA, Opportunistisk, Sensor plug-and-play, Sensorsjälvregistrering, Informationstjänster, Sensorplanering

Innehåll

1 Introduktion **11**
1.1 Bakgrund och Syfte . . . 11
1.2 Vision och koncept . . . 11
1.3 Nyttan . . . 13
1.4 Avgränsningar . . . 14
1.5 Läsarinstruktioner . . . 14

2 Dialog med tänkbara användare **15**
2.1 Workshop 2007 . . . 15
2.2 Workshop 2009 . . . 15
2.3 Slutdemo 2009 . . . 16

3 Relaterade verksamheter **19**
3.1 Liknande system i omvärlden . . . 19
3.1.1 Service Oriented Architecture - SOA . . . 19
3.1.2 NECC . . . 19
3.1.3 Pervasive Computing . . . 20
3.1.4 Service-Oriented Device Architecture (SODA) . . . 21
3.1.5 The ITA Sensor Fabric . . . 21
3.1.6 SitSystByggaren . . . 22
3.2 FOI-projekt . . . 22
3.2.1 TRIS . . . 22
3.2.2 SEMARK . . . 22

4 Arkitektur **25**
4.1 Sensorer i en informationsmodell . . . 25
4.2 REST Gränssnitt . . . 27
4.3 Moduler . . . 29
4.4 Prototyp . . . 30

5 Övriga Verksamheter **35**
5.1 Samarbeten . . . 35
5.1.1 WISENET . . . 35
5.1.2 SM4ALL . . . 36
5.2 Publikationer . . . 36
5.2.1 Sammanfattning av TOppS årsrapport 2007 . . . 37

5.2.2 Artiklel till ICCRTS 09 38
5.3 Examensarbeten 38
5.3.1 Service Composition Workspace 38
5.3.2 Service Composition in Service Oriented Opportunistic Sensor Networks 39
5.3.3 Graphical Query Builder to discover Sensor Information in Opportunistic Sensor Networks 39
5.3.4 Presentation of Sensor Data in Opportunistic Sensor Networks 40

6 Sammanfattning och Diskussion **41**
6.1 Sammanfattning 41
6.2 Framtid 44

Litteraturförteckning **45**

1 Introduktion

1.1 Bakgrund och Syfte

Ledningssystem är beroende av att kontinuerligt förses med aktuell information från sensorer och/eller mänskliga observatörer. Dessa källor övervakar händelser och situationer där det finns resurser att leda för att kunna välja fördelaktiga handlingsalternativ. Sensorinformation kan erhållas från olika källor, alltifrån spaningsradar i flygplatsernas ledningssystem till heterogena, dynamiska och mobila sensorer. Den senare typen av sensorer medför vissa icke-triviala problem som t ex anslutning, automatisk konfiguration, kommunikation, tillgänglighet och frågan om informationens pålitlighet. Ett sätt att hantera dessa problem är att använda sig av tjänstekonceptet. Tjänstekonceptet har varit i fokus inom ledningsområdet under en längre tid. Service Oriented Architectures (SOA) [34] betraktas som en av hörnstenarna inom begreppet Network Enabling Capability (NEC), och ett sätt att migrera från ett mer tätt integrerat stuprörssystem till mer löst kopplade och nätverkbaserade ledningskomponenter. Det senare är ett viktigt mål vad gäller utformning av arkitekturen av moderna ledningssystem. Dessutom öppnar den snabba tillväxten av nätverksanslutna sensorer upp för nya tillämpningar där det blir möjligt att utveckla och använda tjänstebaserade applikationer i större skala. Mångfalden av sensorer kombinerad med heterogena krav gör det tjänstebaserade synsättet till en utmärkt kandidat för utveckling av funktionella och praktiska system för ickespecialiserade slutanvändare. För de flesta slutanvändare är dock sensorer och tillhörande data enbart medel för att uppnå vissa bestämda mål, vilket innebär att tjänsterna måste vara användarvänliga och öppna, samt kunna kombineras i olika konstellationer för att leverera nya tjänster som möter användarens behov.

Projektet, Dynamiskt Tjänsteramverk för Opportunistiska Sensorer och Sensornät (TOppS), är ett treårigt projekt som initierades 2007. Det övergripande målet med projektet har varit att studera och utveckla ett tjänstebaserat ramverk som förbättrar kapacitet och robusthet hos sensorsystem genom att automatisera processer för sensoranslutning, dataanalys och datarepresentation. Sensorer presenteras som tjänster vilka, på ett dynamiskt sätt, kan sättas samman för att lösa olika behov som användarna har.

1.2 Vision och koncept

Visionen med TOppS har varit en ökad användning av befintliga och nya nätverkssensorer genom att skapa ett ramverk, som möjliggör plug & play uppkoppling av sensorer med automatisk konfigurering på ett för användarna transparent sätt. Sensorerna skall vid anslutning anmäla sig själva som tjänster och

publicera information. Dessa tjänster skall sedan kunna utnyttjas av användarna för att på ett flexibelt sätt lösa olika uppgifter och möta olika behov antingen enskilt eller genom att sättas samman i olika kombinationer. TOppS vilar på tre "hörnpelare":

- Sensorer anmäler och konfigurerar sig själva vid anslutning. Anslutningen skall kunna ske "opportunististiskt" då möjlighet ges, då sensorer ofta kan falla från pga exempelvis tillfällig avbrott.

- Operatören har ett transparent gränssnitt mot sensorn och den information som levereras.

- Operatörens behov understöds genom att utnyttja tjänster från en eller flera sensorer.

Ett tjänsteramverk, som realiserar ovanstående vision har ett gränssnitt mot sensorer och ett mot operatören. Sensorgränssnittet måste uppfylla tre grundläggande krav.

För det *första* bör det ha förmåga till plug-and-play, där sensorn konfigureras och ansluts till nätverket automatiskt. För det *andra* måste det kunna garantera kvaliteten på data som levereras. Detta innebär att sensorgränssnittet måste kunna utföra intern kontroll av data och status, och vid eventuella fel meddela detta eller koppla bort sensorn från nätverket. Vidare bör generad datas kvalitet kontinuerligt skattas. Denna förmåga är avgörande för att vanliga slutanvändare skall kunna lita på sensorerna samt kunna bedömma tillförlitligheten hos datat. Det räcker med ett fåtal sensorer som levererar felaktig data för att försämra trovärdigheten för hela ramverket. För det *tredje* skall möjligheten att manipulera både data och sensorer elimineras. För vissa specifika tillämpningar bör data kodas genom att använda kryptografiska nycklar.

För operatören och ur tjänstesynpunkt finns det också ett antal grundläggande förmågor som bör utvecklas, till exempel automatiska larm baserat på en sensor eller kombinationer av sensorer. Detta i sin tur innebär att fusionering av data/information kan underlättas/tillåtas av tjänster på högre nivå. Dessutom bör det vara möjligt för användarna att sätta samman nya (högnivå-) tjänster då nya krav uppstår eller när nya sensorer tillkommer. Det finns också ett behov av kontinuerlig eller tillfällig lagring av data. Med andra ord måste ramverket tillhandahålla dynamiska, flexibla, säkra och pålitliga tjänster.

Tjänsteramverket måste även kunna hantera och presentera ett brett spektrum av applikationer, allt från enkla sensorer till avancerade kombinationer av sensortjänster och i vissa fall även förmågan till realtidssimuleringar samt att strömma data on-line . Ramverket måste därmed kunna hantera en stor mängd olika tjänster som är såväl heterogena som geografiskt fördelade och även tidsmässigt åtskilda.

Vid utvecklingen av ramverket är flera vägar möjliga. Antingen utvecklas ramverket med hjälp av en top-down, en bottom-up eller en kombinerad strate-

gi, där grundläggande sensorfunktioner introduceras och nya funktioner läggs till.

Den första typen, den altruistiska (top-down), är inriktad på utveckling av tjänster och ställer därmed krav på sensorerna för att uppfylla reglerna i det tjänstebaserade ramverket. Fördelarna är att ramverket blir mer dynamiskt och kan utvecklas baserat på slutanvändarnas krav, samtidigt som det kommer att vara relativt enkelt att inkludera nya tjänster. Nackdelen är att ramverket sannolikt kommer att bli komplext, stort, och så småningom svårt att hantera. Dessutom finns det en överhängande risk att några av de utvecklade funktionerna i ramverket förblir oanvända eller är föråldrade vid lanseringen.

I bottom-up-strategin baseras ramverket på sensorernas förmåga. Här är fördelarna att ramverket blir evolutionärt i sin utveckling genom att grundläggande förmågor implementeras först, och sedan utvecklas ramverket till att omfatta mer avancerade funktioner. Detta bör innebära relativt korta utvecklingstider samt enklare anslutning av sensorer till ramverket. Nackdelen är att ramverket inte kommer att kunna hantera komplexa tjänster på grund av inbyggda begränsningar i ett tidigt skede.

En tredje tänkbar utvecklingsväg är att kombinera top-downöch bottom-uppå ett genomtänkt sätt. Då kan fördelar med bägge synsätten uppnås samtidigt som det går att minska nackdelarna.

1.3 Nyttan

Arbetet inom TOppS baseras på tänkta situationer med många små, billiga, heterogena, statiska eller rörliga sensorer som är löst kopplade och inte alltid är tillgängliga. Dessa passar t ex i kaotiska situationer under krig eller katastrofer. I sådana situationer är det begränsat med tid att placera sensorer på optimala platser. Ibland släpps dessa ner från flygplan eller distribueras av soldater till fots eller från fordon. I sådana situationer är det inte möjligt, eller kommer inte att finnas tillräckligt med tid, att manuellt kalibrera sensorerna eller att upprätta kommunikation med ledningsnoder eller andra sensorer. Detta innebär att mycket av detta (kalibrering, konfiguration, kommunikation, etc.) måste göras på ett *opportunistiskt* sätt: när tillfälle ges och på det då bästa sättet i den givna situationen. Här tar vi också hänsyn till operatören och dennes perspektiv. När behov av information uppstår kan operatören inte alltid få veta, eller välja, vilken sensor som levererar denna information, utan det är endast informationen hon/han är intresserad av. Genom att använda TOppS-ramverket:

- Får de tillgängliga sensorerna en högre utnyttjandegrad.
- Kan den insamlade datamängden användas för fler syften.
- Ökar man spårbarhetsgraden i analys och beslutsstöd.

Detta skulle underlätta operatörernas arbete, snabba upp datahanteringsprocessen samt stödja användarcentrerad informationsframtagning vilket slutligen leder till bättre lägesbild.

1.4 Avgränsningar

TOppS-ramverket utgör en generell infrastruktur och brygga mellan sensor, eller sensornätverk, och operatör. Infrastrukturen erbjuder möjlighet att utveckla och lägga till sensorgränssnitt för olika sensortyper, samt applikationer för olika informationsbehov. Det är också möjligt att lägga till olika högnivåtjänster som t ex fusionstjänster mellan sensorer av de typer som råkar bli tillgängliga i en viss situation, ledningstjänster o dyl. Infrastrukturen är däremot inte ett ledningssystem (och/eller sensorsystem), utan den kan anpassas för att stödja olika ledningssystem. Infrastrukturen är inte heller utvecklad för någon specifik tillämpning utan tillämpningen utgörs av de sensorer, tjänster och användarapplikationer som läggs in eller kopplas till infrastrukturen.

1.5 Läsarinstruktioner

Denna rapport består av 6 kapitel. Detta kapitel har givit en bakgrund till projektet, beskrivit vår vision, samt nyttan med projektet och dess avgränsningar. Kapitel 2 (Dialog med tänkta användare) är en kort sammanfattning av de två användarworkshops som genomförts inom ramen för projektet i syfte att skapa en referensgrupp, inrikta projektets innehåll samt få synpunkter på kommande resp. återkoppling på genomfört arbete. Kapitel 3 (Relaterade Verksamheter) sätter TOppS i sitt sammanhang genom en kartläggning av forskningsområden och trender samt ger exempel på närliggande projekt och verksamheter både nationellt och internationellt. Kapitel 4 (Arkitektur) beskriver vår arkitektur samt design och implementation av det tjänstebaserade ramverket. Detta beskrivs ur ett tämligen djuplodande tekniskt perspektiv. I kapitel 5 (Övriga verksamheter) listar vi våra samarbeten och publikationer. Vi beskriver dessutom de examensarbeten som genomförts inom projektet. Kapitel 6 (Sammanfattning) ger en sammanfattning av projektet, våra slutsatser och fortsatt arbete.

2 Dialog med tänkbara användare

För projekt TOppS var det från första början uppenbart att det var viktigt att kunna bringa sig en uppfattning av hur stort behovet kunde vara av ett system som TOppS avser beskriva. Därför har bl.a. ett par workshops hållits, samt en slutpresentation.

2.1 Workshop 2007

I maj 2007 hölls en workshop på FOI Kista med inbjudna från försvaret, räddningsverk, polis och tull. Med dessa tänkbara användare presenterades de idéer som TOppS tagit fram i detta tidiga skede av projektet. Vi diskuterade de problem med hantering av sensorer och annan informationsinsamlingsutrustning man har idag, och hur ett system baserat på en TOppS-liknande arkitektur skulle kunna förbättra situationen. Resultaten, samt de exempelscenarier vi använde under diskussionerna, finns redovisade i kapitel 3 i TOppS årsrapport för 2007 [7]. Sammanfattningsvis konstaterades att man önskade modulär hård- och mjukvara, förmåga att integrera arvssystem i en ny arkitektur, samt automatisk upptäckt och registrering av nya sensorer i ett nätverk. Pålitliga alarmfunktioner sågs som mycket viktiga. För mer detaljerade diskussioner hänvisas till nämnda årsrapport.

2.2 Workshop 2009

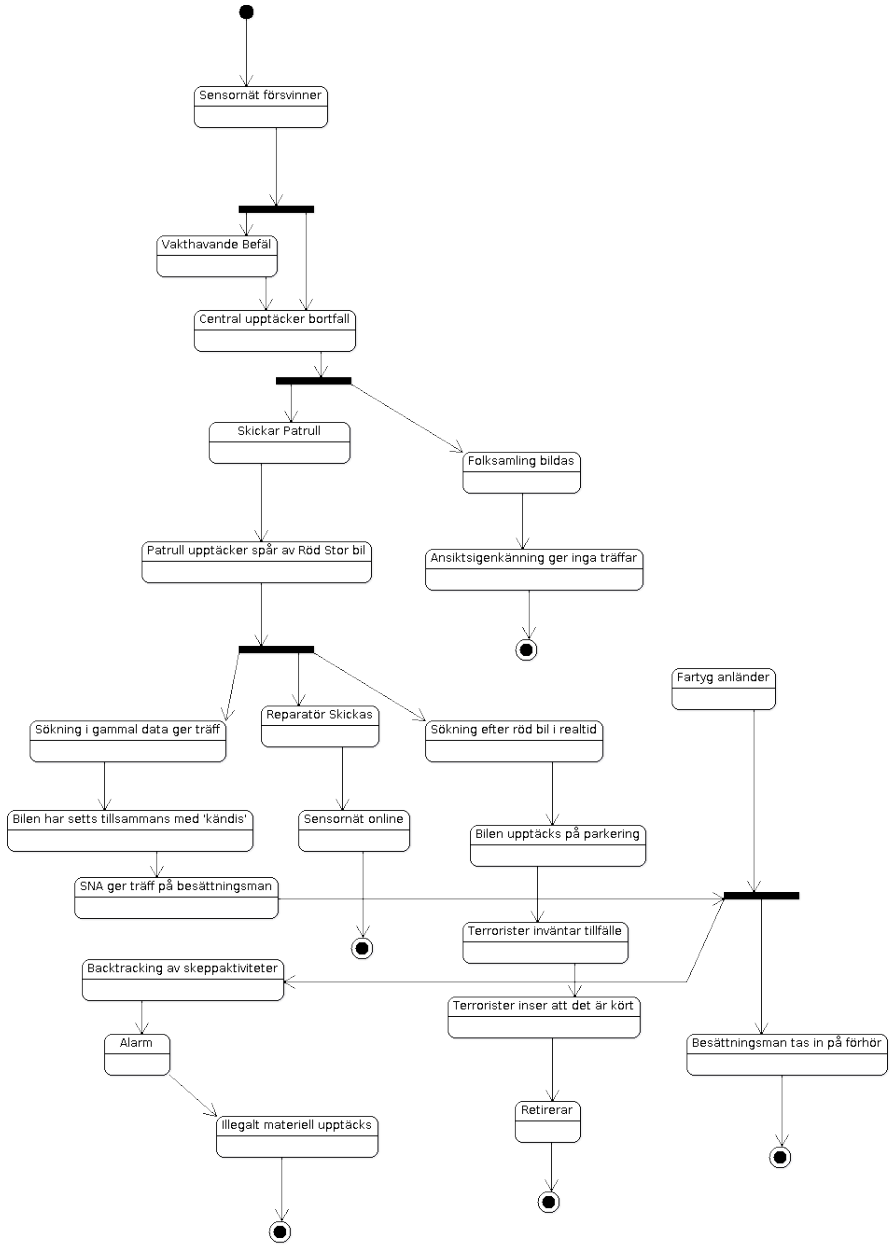
I juni 2009 hölls en andra workshop på FOI Kista med ett mindre antal deltagare. En "brainstorming" session hölls där ett stort antal tänkbara tillämpningsområden för en TOppS-liknande arkitektur föreslogs. Dessa områden grupperades enligt:

- Experiment/Forskningssamverkan
- Dataintegration
- Omvärldsbevakning
- Sökning, datasökning och visualisering
- Behörighetskontroll, övervakning
- Hälsobild, Individövervakning

Workshopen hölls alltför sent under projekttiden för att direkt kunna påverka projektresultatet, men kan komma till nytta vid en eventuell vidareutveckling av TOppS i ett nytt, eller som del av ett mer övergripande projekt.

2.3 Slutdemo 2009

Projektet slutredovisades 1 december 2009 tillsammans med två andra FoT-projekt för en publik bestånde av representanter från försvarsmakten, FMV, FHS, FMKE och FOI. Projektets verksamhet, mål, syfte och resultat presenterades först, varefter scenario och diskussion följde. Prototypens applicerbarhet illustrerades genom användarscenariot vilket utspelades i "Bogaland". Temat var ett smugglingsförsök i BFOR-säkrade Norrköpingshamn. Försöket avvärjs genom en kombination av tjänster såsom information, sensor, social nätverksanalys, skeppsföljning och ansiktsigenkänning; alla ihopkopplade med hjälp av TOppS. Scenariots händelsutveckling redovisas i flödesschemat i Figur 2.1.



Figur 2.1: *Det tänkta händelseflödet som slutpresentationen 1 december 2009 baserades på.*

3 Relaterade verksamheter

I detta kapitel kommer vi att ge en bild av TOppS i relation till andra verksamheter, såväl utanför som inom FOI.

3.1 Liknande system i omvärlden

I omvärlden finns många TOppS-liknande strömningar och tänkesätt varav ett utvalt antal redovisas nedan.

3.1.1 Service Oriented Architecture - SOA

Ett viktigt inslag i TOppS är att sensorer och dataströmmar presenteras som tjänster i stället för fasta trådbundna anslutningar. För att underlätta detta använder vi oss av SOA, som möjliggör tillgång till tjänster utan kunskap om deras underliggande struktur (plattform). SOA har identifierats som en av de teknologier som möjliggör nätverksbaserad ledningsförmåga, och är den arkitektur som används i DoDs NECC och GIG (Global Information Grid) [6]. Tjänsteorienteringen syftar till att dela upp uppgifter, genom att bryta upp ett datorprogram i olika moduler med minsta överlappning i funktionalitet [34], vilket ger en design för snabb och billig systemutveckling och förbättrad systemkvalitet. Webbtjänster (WebServices) är baserade på SOAkonceptet. Webtjänster är plattforms- och språkoberoende, sammansättbara programkomponenter, som är utformade för att möjliggöra interoperabilitet mellan olika applikationer [35, 33]. De stora fördelarna med att använda SOA och webbtjänster är att användarna inte behöver känna till sensortekniska aspekter som arkitektur, kommunikation och implementation. De ger också modularitet och skalbarhet.

3.1.2 NECC

NECC (Net-Enabled Command Capability) beskriver det amerikanska försvarsministeriets (DoD) huvudsakliga ledningsförmåga som har uppgiften att ge befälhavaren den information som behövs för att ta snabba, effektiva och väl underbygga beslut [31]. NECC är baserad på SOA-konceptet och utvecklas löpande och ger nya ledningsfunktioner i en integrerad, interoperabel, samverkande och gemensam lösning. Tanken är att stridskrafter skall snabbt kunna anpassa sig till förändrade uppdragsbehov genom att definiera och anpassa sin information och utnyttja resurser som möjliggör en effektiv och snabb ledning av styrkorna och kontroll över striden. NECC programmet har som syfte att ständigt förbättra ledningsförmågan hos stridskrafterna. Det är baserat på en nätverkscentrerad och tjänstebaserad arkitektur, och tillhandahåller en infrastruktur för beslutsstöd som gör att stridskrafter kan få tillgång till, visa, och förstå den information som behövs för att fatta snabba och effektiva beslut.

Programmet har en användarcentrerad utvecklingsmetodik genom att identifiera förmågebehov, samt utveckla och testa dessa i nära samarbete med användarna. På samma sätt som TOppS har NECC målet att skapa en transparant koppling mellan sensorer och operatörer. Operatören skall få tillgång till den information som den behöver oavsett vilken sensor som leverar den. Det är dock oklart på vilket sätt tjänster presenteras i NECC, hur sammanslagning av tjänster går till och huruvida det erbjuds plug & play funktionalitet för sensorer.

3.1.3 Pervasive Computing

Pervasive Computing är ett framväxande forskningsområde som har fått mycket uppmärksamhet under de senaste decennierna. Pervasive eller "Ubiquitous Computing" (begreppet saknar svensk översättning), som det också kallas, innebär en ny modell för människa-dator interaktion, där datorerna är integrerade i vårt dagliga liv. I detta paradigm kan vi använda flera enheter samtidigt under våra dagliga aktiviteter, ibland utan att ens veta om det. Därmed är datorer inte enbart begränsade till persondatorer, som vi tenderar att tänka på dem, utan också mycket små (även osynliga) apparater. Dessa apparater kan vara mobila eller inbyggda i nästan alla typer av tänkbara föremål, inklusive bilar, verktyg, vitvaror, kläder och olika konsumentvaror, som kommunicerar genom ett sammankopplat nätverk. Sensorer, sensornätverk, trådlös teknik och tjänsteorienterad arkitektur är viktiga begrepp i detta paradigm.

TOppS-visionen bygger delvis på idén om Pervasive Computing och står inför liknande utmaningar. Därför har det varit viktigt för oss att studera detta område och följa utvecklingen. Två särskilda fall av Pervasive Computing är Pervasive Games (PG) och Pervasive Healthcare (PHC). PG [17] är en ny generation av interaktiva spel/distribuerade realtidsapplikationer som kombinerar dataspel med den verkliga världen. Därigenom utvidgar PG spelplanen med miljöer i den fysiska världen. Spelare i dessa typer av spel kan röra sig genom exempelvis en stads gator med mobila enheter (handdatorer, mobiltelefoner o dyl.), interagera med spelet och vara en del av det. Sensorer samlar in information om bl.a. spelarnas positioner när de rör sig. Denna information används sedan för att skapa en spelupplevelse som anpassar sig till spelarnas geografiska position, vad de gör, och även hur de känner. På detta sätt kan spelarna uppleva ett spel som är sammanflätat med den verkliga världen och är potentiellt tillgängligt överallt och när som helst. IPerG är ett EU-finansierat projekt som leds av Swedish Institute of Computer Science (SICS) och bedriver forskning inom både tekniska aspekter och design av PG [29]. Målet med projektet är att utveckla infrastruktur, verktyg och metoder för PG i syfte att bland annat: 1) underlätta en snabb och kostnadseffektiv utveckling av PG, 2) förstå och fånga de potentiella användarnas behov och 3) studera de sociala effekterna av PG.

Pervasive Healthcare [32] syftar till att lösa många av de befintliga och nya utmaningarna inom nuvarande de hälso- och sjukvårdssystemen, såsom

brist på sjukvårdtjänster i många delar av världen, ökande kostnader för dessa tjänster och en ökad nivå av stress. PHC har som vision att leverera sjukvård av hög kvalitet till vem som helst och var som helst med hjälp av ny teknik, t.ex. trådlös kommunikation, för att överbrygga begränsningar kopplade till tid och rum. Liksom i fallet med PG finns många öppna frågor och utmaningar i visionen om PHC. Dessa inkluderar brist på heltäckande trådlösa och mobila nätverk, trådlösa infrastrukturers tillförlitlighet, allmänna begränsningar vad gäller handhållna apparater, medicinsk användbarhet av sensorer och mobila enheter, interferens med andra medicintekniska produkter samt integritet och säkerhet.

3.1.4 Service-Oriented Device Architecture (SODA)

Pervasive Computing kräver en infrastruktur som möjliggör anslutning av olika enheter och integrering av dessa i ett distribuerat system. En sådan infrastruktur är SODA (Service-Oriented Device Architecture), som är en förlängning av SOA (Service-oriented Architecture) paradigmet. SODA bygger på OSGi (Open Services Gateway initiative) [21] och tar denna vision ett steg längre genom att även möjliggöra för enheter (t.ex. sensorer) att presenteras som tjänster [5]. I SODA ansluts enheter till arkitekturen genom så kallade "Device Adapters", som kommunicerar med enheterna och presenterar en abstrakt tjänstemodell av dessa. Därigenom presenterar de enheternas data som tjänster över ett nätverk. SODAs principer och vision är mycket likt dem i TOppS, men det finns inte tillräckligt med detaljerad information för att kunna jämföra de två metoderna. Dessutom är TOppS-begreppet mer omfattande i den mening att det täcker även stegen efter tillhandahållande av tjänster, som omfattar presentation av dessa tjänster och kommunikation med användarna. En annan möjlig skillnad (det är dock inte säkert på grund av bristen på en mer djupgående information om SODA) är förmågan vad gäller plug & play av sensorer i TOppS som saknas i SODA.

3.1.5 The ITA Sensor Fabric

Detta SOA-baserade initiativ [37] har flera interessanta likheter med TOppS. Det utvecklas bl.a. av IBM som en del i "International Technology Alliance in Network and Information Science", ett samarbetsprojekt mellan USA och Storbrittannien. Det är en middleware mellan sensornät och användare som skall hantera sensorupptäckt, registrering, algoritmmatchning till typ av sensorinformation, och anslutning mellan sensorer och användare, allt avsett för ett större nät av sensorer. Även simulerade sensorer kan ingå. Utvecklingsversionen av "Sensor Fabric", se angiven webadress i referensen ovan, har nyligen blivit tillgängling, och kommer alltså litet sent för nyttjande som referens eller liknande till TOppS.

3.1.6 SitSystByggaren

För att kunna skapa situationsanpassade system inom NBF har SitSyst-byggaren framtagits. SitSyst-byggaren [1] kan användas för att skapa situationsanpassade ledningssystem som är tjänstebaserade och distribuerade. Genom att hantera/-konfigurera distribuerade tjänster kan man skapa ett flexibelt system som kan förändras allteftersom förändringar sker.

SitSyst-byggaren (SSB) är en mjukvara som utvecklats för att matcha tjänster och bygga "vyer". Denna vision kan tyckas lik TOppS men SSB är mer specialiserad på att ta fram presentationer än att hantera faktiskt informationslogistik, stödja "Plug 'n' Play" funktionalitet samt kombinera informationskällor. SSB är därtill baserad på CORBA.

3.2 FOI-projekt

På FOI bedrivs några relaterade projekt som är värda att nämnas:

3.2.1 TRIS

Tids/resurskritisk ledning med övervakning av individstatus (TRIS) är ett treårigt FOI-projekt som startade 2008. Syftet är att undersöka möjligheterna att utveckla ett koncept för ett beslutsstödssystem för övervakning av individers status. Övervakningen är tänkt att ske genom insamling av fysiologisk information, omgivningsinformation och eventuellt av rapporter/observationer från olika aktörer. Systemet ska samla in, bearbeta, fusionera och visualisera denna information för att stötta användarens beslutsprocess. Exempel på typiska användare är militära insatsledare eller övningsledare, samt räddningsledare inom akut- och katastrofmedicin.

F-REX (FOI Reconstruction and Exploitation) är ett verktyg för "after action review" som används för visualisering och analys av sensrodata. Det har modifierats inom projektet för att integrera kroppsnätverksdata. Detta arbete är fortfarande i sin linda. Inom projektet har man pss som TOppS samarbetat med WISENET.

TRIS och TOppS förde tidigt samtal om samarbete men då projekten låg i ofas, hann något mer konkret inte utvecklas. För mer information kring TRIS-projektet finns rapporten [14].

3.2.2 SEMARK

SEMARK [8] är ett projekt som under 6 år studerat hur man kan använda ett system bestående av många olika sensortyper monterade på ett antal plattformar. Projektet har arbetat med ett koncept för hur sensorerna bör samarbeta, men man har också bedrivit mätkampanjer med forskningssensorer för att få in underlag att kalibrera olika typer av simuleringsmodeller. Dessa modeller har använts till att att bygga en distribuerad simulator för att simulera sensoran-

vändning vid olika typsituationer med stor noggrannhet.

I SEMARKs konceptstudier har problematiken med planering av sensorinsatser samt styrning och hantering av sensorer och plattformar tillsammans studerats. Detta är även en fråga för TOppS men de båda projekten har tyvärr inte haft möjlighet till närmare samarbete.

Man kan konstatera att tjänsteplanering och kompositionsfunktionerna (som båda projekten berör) är mycket komplexa, både vad gäller hårdvara och logik. Dessutom behöver en helt ny metodik utvecklas för att hantera hela systemet. Grundfunktioner som t.ex navigering och samband måste fungera robust för att hela systemet skall få avsedd verkan.

Plattformar med en eller flera sensorer ombord ska kunna kopplas in och ur, status på samtliga enheter behöver kunna distribueras.

Strukturen på styrfunktionen kan ses som både hierarkisk och nätverksmässig (alla-till-alla) samtidigt beroende på vilket perspektiv man har. Dessutom hanteras tidshorisonter från dagar och veckor ner till millisekunder i extremaste fall. Många användningsfall kräver korta fördröjningar och precis tidskontroll.

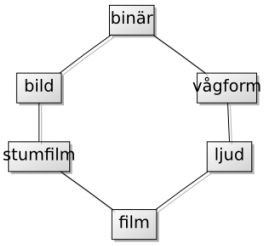
4 Arkitektur

Detta kapitel kommer ge en översiktbild av TOppS-systemet som det fungerar idag utan att djupdyka i tekniska detaljer. Kapitlet är indelat i fyra avsnitt som behandlar typklassning, informationsmodellen, REST gränssnitt, modulerna i systemet samt prototypen.

4.1 Sensorer i en informationsmodell

Innan vi beskriver informationsmodellen bör läsaren uppmärksamma att vi kommer tala om typer eller typklasser och det kan vara bra att först ha en grundläggande förståelse av vad som menas.

Den vanligaste tillämpningen av typteori är i syfte att analysera huruvida metoder är tillämpbara på vissa värden. Detta passar TOppS då tjänster kan ses som funktioner som tar specifika typdata som operatorer.



Figur 4.1: *Det finns många sorter inom typteori. Datatyper är typsättning av värden, klasser är typsättning av objekt och sorter är typsättning av typer*

Typklasser byggs hierarkiskt, ett exempel på en simplistisk modell ges i Figur 4.1. Om vi strikt följer figuren härleder systemet att data vi hittar som är märkt med "stumfilm" även kan hanteras av alla tjänster som fungerar för "bild" och "binär". Exempel på tjänster som är verksamma på en "binär" typ är exempelvis (de)krypteringstjänster och (de)kompressionstjänster. Tjänster baserade på "bild"typ skulle kunna göra ansiktsigenkänning och tjänster för "stumfilm" kan tänkas rita ut projektioner på rörelser.

Man kan utöka resonemanget ytterligare, data skall inte bara produceras utan även konsumeras utav någon eller något. I fallet TOppS handlar det om datavisualisering som blir de primära datakonsumenterna. Vyerna blir alltså typklassade på samma sätt som tjänsterna i systemet.

TOppS baserar sin informationsinhämtning på att sensorer betraktas som

producenttjänster av typsatt data. Denna typsättning gör det möjligt att bygga sensortypklasser vilka därefter kan kopplas ihop med godtyckliga tjänster på ett mer dynamiskt vis. Figur 4.2 visar de grundläggande elementen för TOppS.

Ingång Symboliserar en dataport där data flödar "innåt".

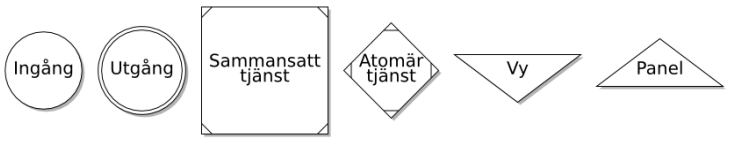
Utgång Symboliserar en dataport där data flödar "utåt".

Sammansatt tjänst Symboliserar komplexa tjänster där vi kan manipulera dataflödet. Dessa har alltid en eller flera ingångar och/eller utgångar.

Atomär tjänst Symboliserar en omanipulerbar beräkning. Dessa har alltid en eller flera ingångar och/eller utgångar.

Vy Symboliserar en datavisualisator specialiserad på en typ. Dessa har en eller flera ingångar.

Panel Symboliserar operatörens kontrollpanel för en given typ (sensor). Dessa har en eller flera utgångar.

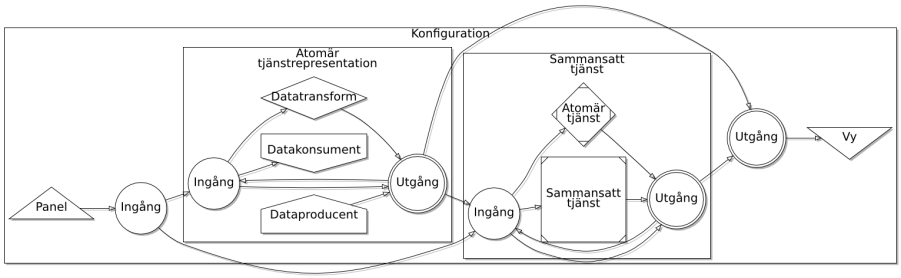


Figur 4.2: *Typinformationen är kopplad till ingångar och utgångar hos tjänster. När man specificerar ett flöde gör man i själva verket detta mellan olika portar och typkontrollen görs således direkt gentemot dessa. Bilden visar de fundamentala typerna i modellen där en sensor representeras av en atomär producenttjänst.*

När sensorer blir sammanlänkade med tjänster och andra sensorer bör de grundläggande typklasserna aggreggeras. Utan denna aggregering tappar man typinformationen och datasäkerheten försvinner, alltså försvinner garantin att operatören får en begriplig vy av datan. Dessa sammanlänkningar behandlas av systemet som en tjänst i sig självt och kan alltså återanvändas så länge deras tjänst- och sensorberoenden fortfarande är vid liv.

Motiveringen för detta är att typklassningen är spårbar genom konfigurationen i vilket fall eftersom data annars inte kan tolkas. En komplex tjänst måste fortfarande presentera ingångar och/eller utgångar då den i sig självt är värdelös. Detta leder till att distinktionen mellan sensor, tjänst och konfiguration suddas ut. Figur 4.3 visar på hur en konfiguration är uppbyggd med hjälp av symboliken från Figur 4.2.

Atomära tjänster kan ej analyseras ytterligare genom dekomposition av systemet utan levereras och hanteras enbart genom något tjänsteavtal. Det är dessa atomära tjänster som levererar mer komplex funktionalitet. Om det inte



Figur 4.3: *Det som skiljer en konfiguration ifrån en komplex tjänst är inte mycket. Bilden är en abstraktion över de möjliga kopplingar som kan representeras. Denna bild visar konfigurationen som ett toppelement data visualiseras genom en vy. Kontroll för en specifik vy görs genom paneler som kopplas till konfigurationen. Konfigurationen består av komplexa tjänster vilka mynnar ut i atomära tjänster. En konfiguration kan kopplas till flera olika vyer och paneler.*

varit tydligt tidigare kan komplexa tjänster bestå av andra komplexa tjänster samt atomära tjänster.

Typkontrollen är baserad på Hindley-Milners typinferens utvidgad med semantisk typdeklaration (alltså skall typerna återfinnas i någon ontologi). För mer information om semantiska inferenssystem se rapport [2].

4.2 REST Gränssnitt

Representational State Transfer eller REST är en infrastruktur med syfte att komma bort från RPC liknande lösningar, som CORBA eller WSDL/SOAP, och skapa löst kopplade infrastrukturer [18, 26].

Centralt för REST är en resursbaserad tillståndslös server-client interaktion. Med detta menas att man vid design av en REST-baserad arkitektur funderar kring vilka resurser systemet skall hushålla (se Tabell 4.1, vilket utbytesformat som gäller för dessa samt i vilka kontexter som resurser anropas. Därtill används de vanliga HTTP verben `GET`, `PUT`, `POST` och `DELETE` för att manipulera resurserna.

Med kontext menar vi, då interaktionen är tillståndslös, att vi genom en URI ger anropet en så tydlig funktion att tillståndet blir meningslöst. Man kan tänka sig att man, analogt med Tabell 4.1, vill få alla användare vilka har en doktorstitel på FOI. Anropet ser då ut som följer: `http://foi.se/user/phD`. För den som vill dyka ned i detta ämne finns RESTwiki [23].

Inom TOppS används resurserna:

`/service/` Visar alla tjänster (sensorer) vilka registerats gentemot systemet. Tjänsten deklarerar vilka portar den är kopplad till, organisatorisk tillhö-

Verb & Resurer	GET	PUT	POST	DELETE
http://foi.se/user	Visar alla användare med deras tillhörande id	Ersätter hela listan med användare	Skapar en ny användare med ett id som genererats av samlingen	Tar bort hela samlingen med användare
http://foi.se/user/edward	Visar specifik användare	Updaterar denna användare eller skapar en ny med givet id	Användaren tolkas som en samling och skapar en undergrupp av användare, tänk användaren http://foi.se/user/phD	Tar bort användaren

Tabell 4.1: *Typiskt REST beteende för en "user" resurs*

righet samt underhållskyldig. Dessa tjänster har även en /log/ URI vilket generar en RSS-ström, över alla metahändelser, samt en /access/ URI vilken beskriver när och av vem anslutningar gjorts. Tjänsten beskriver även sin aggreggerade typ.

/port/ Beskriver riktning på dataporten, typen samt ev. extra steg som behöver tas innan data kan skickas. Innehar en /log/ URI vilken beskriver konsumerad eller producerad data.

/user/ Samlar alla användare som inhämtats till noden.

/key/ X.509 publika nycklar för autentisiering och auktorisering, för både tjänster och användare, samma ID som för respektive egen resurs. Ett exempel är en extern tjänst som skall användas tillsammans med interna. Tjänsten, samt dess kommunikationsparter, tilldelas då certifikat för kommunikation mot ett tillfälligt CA certifikat.

/ontology/ Visar de ontologier som är i bruk. Olika tjänster kan ex. vara typsatta med olika ontologier med reservationen att det finns en tredjeparts ontologi som binder samman dessa.

/node/ Beskriver allt det som är av centralt intresse relaterat till en specifik nod, vilka andra noder den är kopplad till, vilka anslutningar som gjorts och av vem.

`/urimap/` Returnerar alla resurs-URI:er med tillhörande representationer och stödda verb. Exempelvis går det enbart att använda `GET` på `/log/` och då returneras en RSS-representation av loggen.

4.3 Moduler

TOppS är uppbyggt kring en rad tjänster och komponenter vilka presenteras genom sin tjänstebuss och som webbtjänster. Det är i själva verket tjänstebussen som presenterar de kända tjänsterna som webbtjänster och som bygger interna tjänstekontrakt från externa tjänster.

En komplett vy av alla dessa moduler och deras relationer finns återgiven i Figur 4.4. Vi beskriver dock alla dessa i ordning, från operatör till sensor nedan:

Operatörsvy Ger ett webbgränssnitt över hela systemet vilket visas i Figur 4.5. Den är i sin tur uppdelad i:

- **Kartvy** Visar alla tjänster som systemet känner till samt ger möjlighet till filtrering bland dessa. Filter kan appliceras på typ, vilket betyder att man kan välja att exempelvis få se alla värmekameror. Alternativt kan man bygga områdesmönster på kartan och bara se de sensorer som finns inom detta område. Denna går att beskåda i Figur 4.5.a
- **Tjänstorkestreringsverktyget** Kopplar ihop sensorer, tjänster och vyer. Gränssnittet gör det möjligt att bygga de konfigurationer vi tidigare nämnt med hjälp av typkontrollanten. Denna finns att beskåda i Figur 4.5.b.
- **Informationsvy** Visar de vyer och paneler vilka har aktiva konfigurationer. Gränssnittet visar även en logg över de händelser som påverkar de inblandade tjänsterna, främst tillskott och försvinnandet av sensorer. I Figur 4.5.c kallad “Mashup View”

Upptäckare Används för att leta upp tjänster och sensorer som passar in på ett specifikt tjänstkontrakt.

Publicist Används för att registrera funna tjänster i registret och härleder tjänstkontrakt för dessa.

Typkontrollant Är skyldig att analysera typinformation och kontrollera huruvida två typer är kompatibla eller ej.

Presentator Presenterar sensorer som anropsbara tjänster. Presentatorn är den del i systemet som har ansvar för att en sensor avregistreras när den fallit ifrån och återupptäcks när den försöker återansluta. I dagsläge används ett platt strängbaserat kommunikationsprotokoll mellan sensorer och deras specifika presentatorer.

Logg Skriver ner alla händelser tillhörande en körning. Loggen används inte bara för att beskriva metahändelser i systemet, som att sensorer tillkommer eller trillar ifrån, utan är den modul som hanterar lagring av data. Alla loggar i systemet har en egen RSS-ström vilket medför att man kan följa en specifik sensor genom kompatibla läsare.

Tjänstkontrakt för systemets tjänster Tillhandahåller gränsnittsspecifikationerna för alla TOppS-specifika tjänster som skall presenteras. Denna modul måste således installeras på alla deltagande maskiner.

Sensorsimulator Modulen kan presentera tjänster till systemet i utvärderingsyfte.

Enhetsstartare Enhetsberoende del vilken kopplar till och kör igång (dvs. "bootstrappar") sensorn in i TOppS. Enhetshanteraren kan verka som en brygga och synkront reläa data samt kommandon mellan sensor och tjänster. Grundtanken är dock att denna enbart är verksam ett kort tag innan sensor och Presentator lyckats komma överens. En sinnebild för denna del kan vara `autostart.inf` stödet i Windows eller `udev` subsystemet i Ubuntu.

4.4 Prototyp

I TOppS har vi utvecklat en prototyp i syfte att demonstrera och utvärdera konceptet. Figur 4.5 visar bilder från webbgränssnittet som utvecklats för TOppS.

De teknikval som gjorts för prototypen beskrivs här:

CXF REST Vi väljer att använda Apache CXF:s implementation av REST [4] standarden.

OSGi Står för Open Service Gateway [20] Interface och är ett förkroppsligande av tankarna kring SOA [25]. Mer specifikt är OSGi en specifikation för hur utvecklare av bytekod enklare skall kunna skapa modulariserad kod, i detta fall exporteras applikationsfunktionalitet i form av tjänster genom gränssnittsspecifikationer.

R-OSGi I dagsläget används en lättviktstransportsystem för distribution av OSGi tjänster. Vi planerar dock att skrota R-OSGi till fördel för Apache CXF:s D-OSGi [3] vilken stödjer betydligt fler transportprotokoll.

Spring Är en konfigurationsteknik [27] för bytekod. Mycket kod som skrivs hanterar specifika instansfall vilket eg. kan argumenteras vara konfigurationspecifik för en specifik körmiljö. Att använda en teknik som Spring innebär en generaliserad kodbas med mindre "lim".

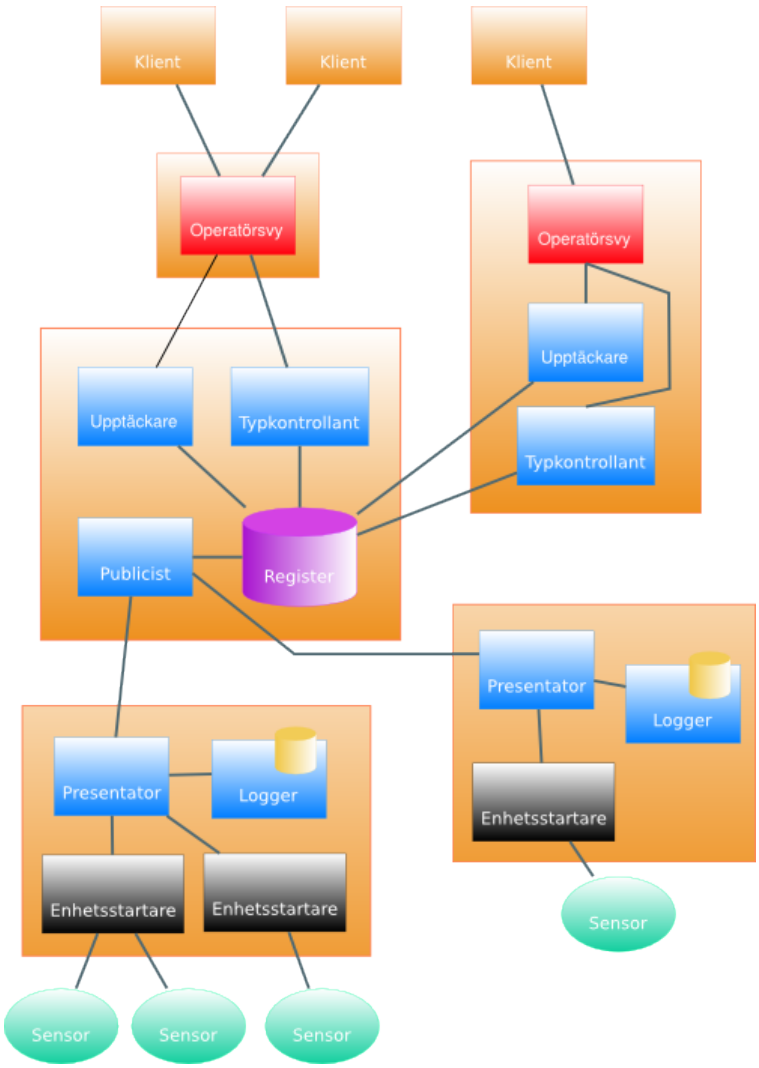
Hibernate Är ett system för databasabstraktion för bytekod [10]. Detta ger möjligheten att peka om vilken databas som applikationen använder utan att behöva skriva ett eget konfigurationssystem eller omkompilering av koden. Kombinationen av Spring och Hibernate är särskilt lyckad.

dmserver Som referensmiljö använder vi tre noder vilka kör Spring dmserver [28] där modulerna är spridda och/eller replikerade över noder.

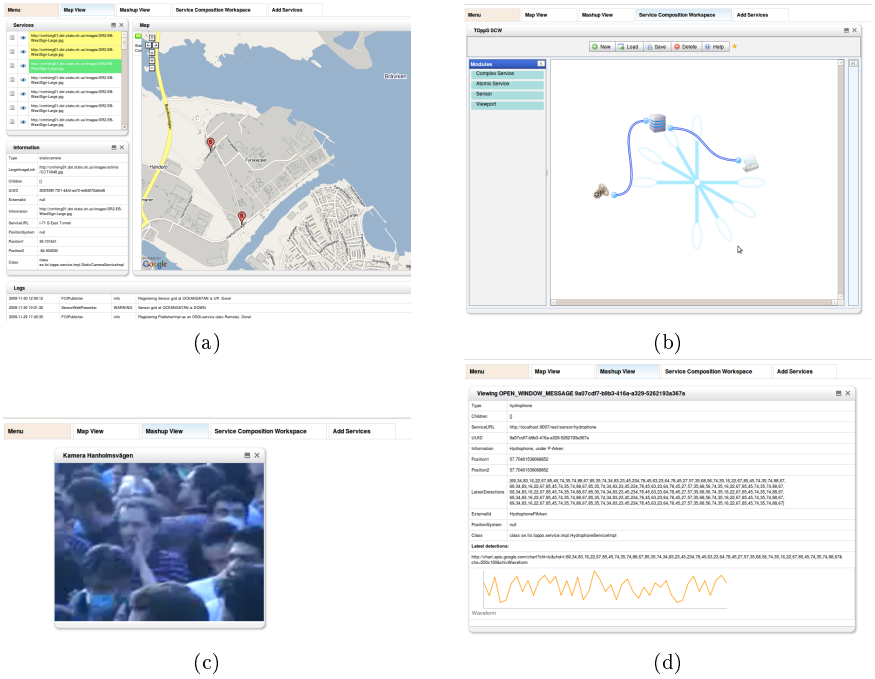
Maven Ett byggsystem med stöd för inladdning av transitiva beroenden.

PAX OPS4J [19] initiativet har mynnat ut i, bland annat, PAX [22] för att underlätta byggandet av OSGi moduler. Verktyget körs genom maven.

Java I dagsläget används Java för TOppS-specifik bytekod. Vi tittar dock även på Scala [30] i syfte att förenkla kodbasen.



Figur 4.4: *Ett exempel på hur ett TOppS system skulle kunna tänkas se ut. Alla moduler är enkla att flytta på så en alternativ systemkonfiguration skulle kunna vara att ha alla tjänster inklusive webbgränssnitt och enhetsstartare på samma maskin. De orangea kvadraterna representerar olika maskiner för vilka TOppS moduler har startats.*



(a) (b)

(c) (d)

Figur 4.5: *(a) TOppS-översiktsbild som visar de sensorer systemet känner till. Markerade sensorer visas på karta samt det som är känt om den valda sensorn. Längst ner syns logg-fönstret där meddelanden om olika händelser visas. Dessa meddelanden skickas även som en RSS-ström och kan prenumereras på med valfri RSS-läsare.*
(b) Bild på tjänstsammansättningsverktyget som hjälper användaren att koppla ihop sensor och / eller tjänst med tillgänglig vy. Man kan tänka sig att detta fall visar en bildsensor som matar data till en bildanalystjänst för ansiktsigenkänning och slutligen att datat presenteras på en vy som klarar av att visa bilder.
(c) Bilden visar Mash-up vyn specificerarad av en konfiguration (tex. den beskrivna i bild c) med ett instantierat bildvisningsfönster. Kontrollströmmar kan ej skickas genom vyn utan kräver att man specificerat paneler i en konfiguration genom tjänstesammansättningsverktyget.
(d) Bilden visar exempel på inspektion av en specifik hydrofondetektion (med emulerad data) i Mash-up vyn.

5 Övriga Verksamheter

5.1 Samarbeten

Då projektets verksamhet berör ett brett område som sträcker sig från sensorer till tjänster har ett spår varit att försöka samarbeta med andra projekt och verksamheter inom berörda områden. Detta för att så långt som möjligt kunna dra nytta av andras domänkunskap samt erfarenhet. De två samarbeten som projektet kom längst med har varit WISENET och SM4All. Dessa beskrivs i följande avsnitt.

5.1.1 WISENET

WISENET är ett Vinn Excellence Center sponsrat av Vinnova, Universitets- och Industripartners[1] [36]. Det kommer att, uppdelat på två faser, vara verksamt under åren 2007-2017. Syftet med detta center är att studera trådlösa sensornätverk (Wireless Sensor Networks). Visionen för centret är att trådlösa sensorer och sensornätverk, inom en snar framtid, skall kunna göras så små, billiga, energieffektiva och smarta att de kommer att kunna få en mängd tillämpningområden i vardagen samt inom samhälle och industri. Dessa trådlösa sensorer kommer m.h.a. effektiva kommunikationsprotokoll att kunna samlas i stora[2] nät som snabbt kan läggas ut och initieras.

Forskningen inom centret har inriktats mot:

- Hur man kan integrera avläsning, data processning och kommunikation i en sensorenhet.
- Att hantera energiförbrukning samt att kunna generera ny energi.
- Att skapa sensornätverk som är självkonfigurerande, robusta och underhållsfria under en period på upp till 10 år.
- Att kunna koppla sensorer till Internet på ett säkert sätt.
- Att reducera storleken på sensorer samt tillverkningskostnaden.

[1] Parterna i detta center har varit många och förutom FOI har parterna i fas 1 varit: Banverket `http://www.banverket.se`, Hectronic AB `http://www.hectronic.se`, SenseAir AB `http://www.senseair.se`, SICS (Swedish Institute of Computer Science AB) `http://www.sics.se`, Silex Microsystems AB `http://www.silexmicrosystems.com`, TRIONA AB `http://www.triona.se`, TNT-Elektronik AB `http://www.radiokommunikation.se`, Uppsala Universitet `http://www.wisenet.uu.se` samt WiseNet Holding AB. Under fas 2 kommer att tillkomma ytterligare partners, men huvudaktörerna är fortfarande Uppsala Universitet samt SICS.

[2] Sensornät innehållandes hundratals och kanske tusentals sensorer.

TOppS-projektets samarbete med WISENET har framförallt bestått i att delta i en arbetsgrupp vars syfte har varit att enkelt kunna koppla sensorer till Internet samt att kunna presentera dem som tjänster.

5.1.2 SM4ALL

SM4ALL (*Smart hoMes for All*) är ett EU-projekt inom FP7 med målet att skapa ett hem som med hjälp av smarta tjänster gör vardagen lättare [24]. Tanken är att detta hem skall kunna användas av såväl normalfungerande personer som handikappade. Projektet består av tio parter från fem länder[3] som kommer från både högskola och industri samt är av varierande storlek.

Projektet kommer att skapa ett innovativt ramverk för smarta inbyggda tjänster i en inbäddad (*pervasive* och *immersive*) och person-centrisk miljö. Detta skall ske med hjälp av bl.a. semantiska metoder och tekniker för dynamisk tjänstekonfiguration. Vidare kommer det att, förutom de mer traditionella, finnas hjärnstyrda gränssnitt (s.k. BCI, $BrainComputerInterface$) som kommer att vara speciellt utformade för förlamade personer.

Det finns för närvarande två scenarier som projektet arbetar med. Det första handlar om en nästan förlamad individ som skall kunna kontrollera sin miljö (ex. ljuset skall släckas när han vill sova, temperaturen skall regleras enligt önskemål, ytterdörren skall automatiskt låsas), beställa mat och andra tjänster samt kunna kommunicera med andra. Det andra scenariot gäller en normalfungerande person som skall ha en fest.

Exempel på de forskningsfrågor som tas upp i projektet är:

- Användargränssnitt för olika typer av personer
- Semantiska tjänstebeskrivningar, sammansättningar och hantering
- Ad-hoc sensorhantering med automatisk konfiguration och felhantering

Samarbetet mellan TOppS och SM4All har främst handlat om de semantiska tjänstebeskrivningarna samt interoperabilitetfrågor mellan de ingående ramverkskomponenterna i SM4All arkitekturen.

5.2 Publikationer

I projektet har det publicerats en förstudierapport samt en konferensartikel. I följande avsnitt återfinns kortare sammanfattningar av dessa publikationer.

[3]Partners i konsortiet är, förutom FOI, SAPIENZA Universita di Roma `http://www.uniroma1.it/default_e.php`, Elsag Datamaturl `http://www.elsagdatamat.com/EN/Home.htm`, Technische Universitaet Wien `http://www.tuwien.ac.at/tu_vienna`, Fondazione Santa Lucia `http://www.hsantalucia.it`, Guger Technologies `http://www.gtec.at`, University of Groningen `http://www.cs.rug.nl/ds`, Thuiszorg Het Friese Land `http://www.thfl.nl`, Telefonica Investigacion y Desarrollo `http://www.tid.es/en` och Kungliga Tekniska Högskolan `http://www.kth.se/?l=en_UK`.

5.2.1 Sammanfattning av TOppS årsrapport 2007

Ett grundläggande problem vad gäller sensorer är tillgängligheten, tillgång till teknisk information om sensorerna som sådana samt metainformation om levererad sensordata, såsom använd datamodell etc. Många användare av data känner en stor osäkerhet i att utnyttja "andras" sensorer med följden att data inte används i den utsträckning som sensorägaren tänkt sig. Genom att förvandla den traditionella sensorn till en nätverksnod kan tillgängligheten väsentligen förbättras, men det krävs att nya protokoll och tjänster utvecklas liksom att sensorn tillförs nya egenskaper. TOppS projektet har studerat problematiken och genomfört en fördjupad analys av tekniken både från ett tekniskt perspektiv och från användarens synvinkel. Resultaten publicerades i TOppS årsrapport för 2007[7]. Utgångspunkten i analysen var opportunistiska sensornätverk, det vill säga nätverk där sensorer kommer och går på ett för mottagaren ofta okontrollerat sätt. En följd av denna dynamik är att de tjänster som utvecklas klarar av ett *ad-hoc* sammansatt sensornätverk.

Tillämpningar har identifierats, däribland övervakning av personers fysiska status samt övervakning av skyddsvärda områden, för vilka det finns både militära och civila exempel. För en pluton kan den fysiska statusen övervakas, och med samma teknik kan personlig hälsoövervakning genomföras i hemmet. Hamnar och förläggningar (Camps) är ett annat exempel på "dual-use".

Vid framtagandet av sensorbaserade tjänster står vi inför följande utmaningar: Att undvika hög komplexitet som ofta uppkommer som en följd av heterogena sensorstrukturer, vilket får till följd att lättanvänd blir obrukbar. Att lösa ägarfrågan samt hitta lösningar för informationsintegriteten. Detta för att tillse att rätt användare kommer åt rätt tjänster samt att efterfrågad information hamnar hos respektive slutanvändare. Att leverera tjänster och data med hög datasäkerhet och slutligen att få en kommersiell förankring.

Det bedrivs utveckling och forskning inom området, dock med olika fokus. TOppS projektet har studerat de mest relevanta initiativen och konstaterat att de ofta har samma utgångspunkt men att de skiljer sig i slutmål. Forskningen kan delas in i tre huvudgrupper; tjänsteutveckling, utveckling av mellanprogramvara samt utveckling av sensornära standarder och protokoll. Ett intressant initiativ är Java Distributed Data Acquisition and Control (JDAC), som är en plattform tänkt för att konstruera sensornätverk baserat på Java skript. Ett annat är GRID som har som mål att utnyttja distribuerade datorer för parallella beräkningar. Andra mer sensornära tillämpningar är Sensor Web Enablement (SWE) och Sensor Model Language (SensorML), båda har som syfte att göra sensorer lättanvända i nätverk.

Vilket redan berörts ovan har TOppS projektet identifierat ett antal militära tillämpningar, i vilka nätverksbaserade tjänster kan spela en viktig roll. Kroppsmonterade sensorer för prestationsövervakning av soldater är en viktig tillämpning med stor potential, likaså temporär områdesövervakning av specifika områden. I dessa två exempel kommer sensorer att "komma och gå" och som en följd därav blir även tjänsteinnehållet dynamiskt. För att förankra projek-

tidéer har ett antal möten ägt rum med militära och civila aktörer. De koncept och slutsatser som presenteras i slutrapporten för 2007 är till stor del baserade på dessa möten.

5.2.2 Artiklel till ICCRTS 09

Inom projektet har det skrivits en artikel som publicerades på ledningssystemskonferensen ICCRTS09 [16]. Målet med artikeln var att presentera TOppS som en möjlig infrastruktur att användas i "agila ledningssystem", vilket var årets tema för konferensen. Artikeln var i stor utsträckning en anpassning av årsrapporten 2007 för ICCRTS-forat, samt kompletterad med resultaten från 2008. Den vanligaste lednigssystemsensorn många tänker på är den stora flygledningsradarn vid flygplatser. Men ett ledningssystem som skall klara många heterogena sensorer som har varierande tillförlitlighet och anslutbarhet måste ha ett annat sätt att hantera och kommunicera med sina sensorer. Detta, som ju kan vara fallet i många militära situationer såväl som för räddningsoperationer i svår terräng, belyses i början av artikeln som utmaningar för ledningssystemet. TOppS vision beskrivs, och likaså angränsande aktiviteter runtom i världen, vissa av dem adresserade även i denna slutrapport. Resultatet från den första workshopen 2007 gås igenom i ljuset av två scenarier med personburna sensorer: sådana sensorer för avkänning av bärarens omgivning, respektive av bärarens egen hälsostatus. Därefter redovisas några utmaningar ett ramverk som TOppS skall kunna hantera, varpå TOppS arkitektur beskrivs mera generellt, ej så detaljerat som är fallet i denna slutrapport.

5.3 Examensarbeten

Inom projektet har det genomförts fyra examensarbeten som har utforskat relevanta frågeställningar för TOppS. De har fokuserat på tjänstesammansättning (både användarstyrd samt automatisk), Grafisk frågegenerering av semantiska frågor samt automatisk sensoruppkoppling. De beskrivs nedan under rubriker i form av examensarbetenas engelska titlar.

5.3.1 Service Composition Workspace

En viktig del inom TOppS är möjliggörande av olika systems möjlighet till datautväxling. "Service Composition Workspace", eller Tjänstorkestreringsverktyget, var ett examensarbete som utfördes av Yefeng Liu [15].

Målet var att skapa ett verktyg med vilket man skulle kunna koppla ihop olika typer av tjänster, aggregera dem i mer komplexa tjänster samt baserat på in- och utdata kunna avgöra huruvida tjänsterna var kompatibla. Verktyget är tänkt att generalisera informationaggreggering och tjänsteorkestrering så att operatörvy, fusiontjänster och sensorer får en tät och säker integrering. Resultatet av detta examensarbete är det som ligger till grund för det nuvarande tjänstesammansättningsverktyget, tidigare beskrivet i Arkitekturkapitlet.

5.3.2 Service Composition in Service Oriented Opportunistic Sensor Networks

Målet med detta examensarbete var att, givet en SPARQL-fråga, försöka identifiera de tjänster som krävs för att satisfiera frågan. I de fall där flera typer av tjänster kunde användas för samma funktion gjordes en utvärdering av tjänsteförslagen och sammansättningen av dem. Som kriterium för att avgöra vilken sammansättning som var den bästa användes den sammansättning som krävde minst antal tjänster.

Ponera att användaren vill ha en alarmtjänst som skall kunna varna för eventuella inkräktare. Tjänsten skall kunna täcka ett geografiskt område x och vid detektion kunna visa bildresultatet från en kamera som täcker detta område. Givet är även att vi har en ontologi som beskriver olika typer av resurser såsom sensorer (kameror, akustiska, sesmiska och magnetiska sensorer m.fl.). Deras egenskaper såsom sensormodell, täckningsområde, känslighet, operationsparametrar, o.s.v. är också beskrivna. Ontologin skulle kunna visualiseras som en graf där resurser och deras egenskaper utgör noderna samt relationen till egenskaperna (och varandra) utgör kanterna.

Det första vi gör när frågan kommer är att, genom att traversera ontologigrafen, undersöka vilka resurser som täcker område x. Som svar får vi att vi har en temperatursensor, en rörelsedetektor, en kamera och en IR-kamera. Sedan undersöker vi om kamerorna har möjligheten att detektera potentiella inkräktare. Då ingen av kamerorna har den egenskapen tittar vi på övriga resurser och upptäcker att rörelsedetektorn är en kandidat. Vi har nu två potentiella sätt att lösa frågan. Den efterföljande utvärderingen ger båda lösningar lika mycket poäng (då vi bara tittade på antalet ingående tjänster) men om frågan hade varit mer specifikt ställd, t.ex. att kamerabilderna skulle kunna ses både dag och natt skulle lösningen med IR-kameran fått fler poäng.

Implementationen gjordes i Java m.h.a. Jena. Examensarbetet utfördes av KTH-studenten Andreas Hemmingsson [9].

5.3.3 Graphical Query Builder to discover Sensor Information in Opportunistic Sensor Networks

Målet med detta examensarbete var att underlätta för användaren att ställa en välformulerad semantisk fråga i SPARQL-format. För de flesta användare är inte SPARQL det uppenbara eller lätta sättet att formulera ett tjänstebehov (exempelvis alarmtjänsten vi diskuterade i det förra examensarbetet). Lösningen som undersöktes var att istället låta användaren rita upp en graf som sedan översattes till SPARQL av systemet.

Användaren har en vy där hon/han ser de resurser och egenskaper som är tillåtna att användas. Denna lista bygger på samma ontologi som i det tidigare examensarbetet. Man skulle kunna säga att frågan uttryckt som graf egentligen utgör en delmängd av ontologin men med instantierade noder. Det som sker då

användaren vill ställa en fråga är att hon/han börjar med att välja ett begrepp, exempelvis en kamera. Givet kameradefinitionen kan användaren sedan välja de egenskaper som kameran skall ha, t.ex. placering. När användaren var klar med att bygga sin graf översatte systemet automatiskt grafen till SPARQL.

Problemen med detta angreppsätt till frågebyggande var att användaren är tvungen att ha hela bilden över ontologins struktur klar för sig samt att frågan i grafisk form faktiskt inte är så mycket enklare att konstruera än om användaren hade försökt göra det direkt i SPARQL.

Implementationen gjordes i Java m.h.a. Jena, ett semantiskt webramverk för Java [11], av KTH-studenten A.F.M Sultanul Kabir [13].

5.3.4 Presentation of Sensor Data in Opportunistic Sensor Networks

Ett av målen med TOppS är att ha dynamisk inkoppling och konfigurering av sensorer. Detta examensarbetes mål var att undersöka och försöka åstadkomma detta. Då implementationen utnyttjade UDEV-funktionaliteten gjordes implementationen i Linux-miljö.

Examensarbetaren skapade ett antal UDEVregler som vid inkoppling av en USB enhet till en dator automatiskt detekterade sensortyp. Givet att en viss sensor detekterats skapas sedan en webtjänst som presenterar sensorn för omvärlden.

Tillvägagångssättet fungerade bra för mycket enkla sensorer. Det som gjordes var att låta ett USB-minne emulera olika typer av sensorer. Tyvärr lyckades han inte att få det att fungera med en något mer avanserad sensor som t.ex. en USBkamera.

Examensarbetaren var Pradeep Jonnaganti och hans arbete finns att läsa i [12].

6 Sammanfattning och Diskussion

I detta kapitel diskuteras och sammanfattas projektets verksamhet samt de resultat som åstadkommits. Vi kommer även att diskutera rekommendationer för framtiden samt möjliga fortsättningsspår för projektet.

6.1 Sammanfattning

Projektets mål under de tre verksamhetsåren har varit att studera och utveckla ett tjänsteramverk med syfte att lyfta sensorer och hanteringen av dem till en mer abstraherad nivå. Det vill säga ett nytt synsätt där sensorer, ur användarsynpunkt, betraktas som vilken annan tjänst som helst.

Detta avsnitt är uppdelat i tre delar. I den första går vi igenom utmaningarna som TOppS tjänstesynsätt innebär, diskuterar varför de är utmaningar samt vilka TOppS-projektektet adresserat. I den andra delen tar vi upp de vinster som TOppS synsätt medför och slutligen tar vi upp de resultat som projektet åstadkommit under de tre verksamhetsåren.

TOppS tjänstesynsätt medför ett antal utmaningar som måste övervinnas innan man når ända fram. Exempel på utmaningar:

- *Hur kan vi få en tvåvägskommunikation med sensorer där de ansluter och anmäler sig själva till systemet samt tillåter oss att styra och konfigurera dem genom det distribuerade ramverket?* Denna frågeställning är egentligen tvådelad då det handlar om kommunikationen *till* och *från* sensorn. Om vi funderar på det första fallet, d.v.s. kommunikation till sensorn är det i stort sätt beroende av vilken typ och ”intelligens” som återfinns hos sensorn. Då TOppS inte haft som fokus att utveckla eller föreslå nya sensortyper har vi nöjt oss med att konstatera att de flesta sensorer ur detta hänseende är passiva och inte tillåter någon konfiguring utifrån. Vi tror oss dock se trenden att sensorer blir allt mer dynamiska och intelligenta. Det andra fallet, kommunikation från sensorn, har varit ett krav vid designen för TOppS ramverk. Här kan man konstatera att det finns många olika protokoll och standarder att ta hänsyn till. Lösningen har varit att skapa ett gränssnitt för vilket man kan skapa externa fasader anpassade gentemot specifikt önskad standard. För en djupare diskussion se kapitel 4.

- *Hur hanterar vi heterogena sensorer?* Då vi inte ville begränsa systemet till någon enkild sensortyp, har detta också varit ett designkrav för TOppS-ramverket. På samma sätt som ovan har vi löst det med det generiska gränssnittet vilket ger en unifierad sensorpresentation för TOppS-systemet.

- *Hur modellerar vi något så heterogent som informationstjänster på nätet*

som tjänster, och hur kopplar vi tjänster till varandra? Då ingen idag beskriver heterogena tjänster på ett koherent vis måste vi skapa en egen representation för att p.s.s. få ett slutet beskrivningssystem. Vi tar frågan kring informationsproduktion och -konsumption till ett funktionellt typresonemang där tjänster interagerar genom kompatibla typsättningar. De faktiska kopplingarna görs därefter typsäkert genom att användaren, på ett understött sätt, "kopplar" ihop olika tjänster genom tjänsteorkestreringsverktyget. Med understött menar vi att användaren får felmeddelande och förbjuds koppla ihop två tjänster vars typer ej matchar.

- *Hur hanterar vi belastning på tjänsterna?* Avlastning, eller lastbalansering, sker alltid genom varje sensors respektive Presentator. Observera att data läses från dessa presentatorer och inte från sensorn direkt. I fallet sensorkontroll blir det lite mer invecklat. Men detta är ett problem vi inte har tacklat och återstår för framtiden.

- *Hur hanterar vi osäkerhet och tillförlitlighet?* En sensorbaserad tjänst har naturligt en viss osäkerhet i det data som produceras. Vidare kan tillförlitlighet även vara beroende av faktorer som underhåll, kalibrering och ägare. För att TOppS-ramverket skall kunna hantera dessa faktorer krävs att sensorn själv kan meddela detta eller att en auktoriserad användare kan lägga till den informationen separat.

- *Hur autentisierar och auktoriserar vi sensorer och användare i en distribuerad miljö?* Det går att bygga informationskretsar i systemet som inte är autentisierade eller auktoriserade. För varje "säker" informationskrets utbyts nycklar vilka är godkända av en temporär certifikat auktoritet (CA). Informationsnivån bestäms utefter den sensor vars roll är högst klassad. Information från lägre informationsnivåer tillåts passera uppåt men ej åt andra hållet.

- *Hur sker datadistribution när det kommer till långtidslagring och återinhämtning, dvs sökning i gammal data?* Loggningsfunktionaliteten i TOppS är inte satt i sten, men ett typiskt scenario blir att systemet lagrar producerad information utefter roll samt krypterat med nodens långtidscertifikat. Utchecking av data sker därefter auktoriserat utefter rollen för användaren. Detta är i dagsläget ännu ej implementerat. Annotation av producerad bild- och ljuddata är inget vi har fokuserat på i TOppS utan ser på detta som en tjänst man kopplar in till ramverket.

- *Hur behåller vi datats relationer så att vi kan spela upp händelseförlopp i analyssyfte?* I en sammansatt tjänst kan det ingå ett flertal datagenererande, -filtrerande och -aggregerande tjänster. TOppS har för närvarande endast hunnit titta på hur man kan lagra data som kommer direkt från en datagenererande tjänst. Att se till att relationer mellan olika data kan behållas för att spela upp händelseförlopp återstår som ett framtida arbete.

I dagsläget fungerar detta genom att:

1. Den komplexa tjänsten skapas, certifikat utdelas och datakretsen auktoriseras (d.v.s. om den inte tillhör en "osäker" kanal).
2. Den komplexa tjänsten och dess ev. nycklar lagras.
3. En auktoriserad användare söker efter tjänstens data.
4. Användaren erhåller den aggregerade mellanlagrade datan för den sammansatta tjänsten.

TOppS tjänstesynsätt medför även ett antal stora vinster. Några exempel på sådana är:

- Tjänster kan skräddarsys efter användarens behov.
- Transparensen (avseende sensortyper, placering, antal och arbetssätt) för slutanvändaren ökar.
- Tillgängliga sensorer får högre utnyttjandegrad då deras data lagras och kan återanvändas. Vidare kan fler användare m.h.a. ramverket utnyttja en och samma sensor.
- Insamlad data kan användas för fler syften.
- Robustheten ökar i systemet (då en tjänst/sensor enklare kan bytas mot en annan likvärdig vid fel eller överbelastning).
- Spårbarhetsgraden i analys och beslutsstöd ökar (då man lagrar data i systemet och även gör den enklare sökbar).

Många av utmaningarna har projektet lyckats lösa men en del återstår ännu. De resultat som projektet åstadkommit under de tre åren är flera och här följer några exempel:

- Projektet har genomfört en förstudie och studerat de trender som finns både på teknik, metod och idé-sidan. En del av detta återges i kapitel 3 samt i kapitel 5 där förstudierapporten [7] samt artikeln [16] ingår.
- Projektet har designat och utvecklat ett tjänsteramverk som kan hantera en distribuerad, heterogen och dynamisk miljö med både egna och externa tjänster (både sensorbaserade och andra).
- Tjänsteramverket har gjorts så flexibelt att det kan hantera heterogena sensorer och informationstjänster.
- Ramverket tillåter att man har och använder tjänster som ägs och distribueras av olika parter.
- Ramverket har autentisiering av användare (snart kommer även auktorisering att finnas).

- TOppS-ramverket kan distribueras ut på valfritt antal noder. Dessa noder kan sedan säkras upp så att man har hemliga och öppna delar.
- Vid händelser såsom exempelvis att nya tjänster tillkommit eller fallit bort, samt att någon slags information producerats t.ex. att en sensor detekterat något, så skickas meddelanden till användaren.
- Tjänster kan kopplas ihop till sammansatta tjänster. Dock är denna process ännu ej helt automatiserad.
- Tjänster kan sökas och grupperas baserat på metadata såsom typ eller geografiskt område.
- Kontinuerlig kontakt med användare har upprätthållits genom Workshops samt demonstration av TOppS.

Det ramverk och metoder som byggts upp är baserade på flera beprövade tekniker och metoder. Det finns fortfarande delar att arbeta vidare på och det är på intet sätt en färdig applikation. Men, som vi ser det, finns det stor potential med TOppS och vi har lyckats lösa flera av de svåra utmaningarna.

6.2 Framtid

Då vi tror på systemet, och de vinster som finns att få, är vår vision för framtiden att TOppS skall kunna användas till att koppla hela kedjan från data till lägesbild för beslutstöd. Vi ser att systemet skall kunna användas för att enklare planera, beställa och kombinera tjänster. Vidare skall man kunna underlätta för användaren att utnyttja fler sensortjänster genom att enklare kunna anpassa de underliggande sensorernas konfigurationer efter behov.

Litteraturförteckning

[1] Jens Alfredson, Jenny Lindoff, and Carin Rencrantz. Behovsanalys av SitSyst-byggaren - en metodrapport som stöd för den fortsatta utvecklingen av tjänstehanteringsverktyg. Metodrapport FOI-R–1231–SE, FOI, May 2004.

[2] Hirad Asadi, Marianela García Lozano, Andreas Horndahl, Edward Tjörnhammar, and KTH Katharina Rasch. Introduction to Technologies and Methods for Semantic Information Management. User report FOI-R–2858–SE, FOI, December 2009.

[3] CXF Distributed OSGi. `http://cxf.apache.org/distributed-osgi.html`. (senast besökt 03 december 2009).

[4] CXF RESTful Services. `http://cxf.apache.org/docs/restful-services.html`. (senast besökt 03 december 2009).

[5] S. de Deugd, R. Carroll, K. Kelly, B. Millett, and J. Ricker. Soda: Service oriented device architecture. *IEEE Pervasive Computing*, 5(3), 2006.

[6] Department of Defense Global Information Grid Architectural Vision. `http://www.defenselink.mil/cio-nii/docs/GIGArchVision.pdf`. (senast besökt 10 november 2009).

[7] T. Fristedt, M. Garcia Lozano, P. Hörling, F. Moradi, J. Pelo, and P. Sigray. Service Oriented Opportunistic Sensor Networks (TOppS) - Feasibility Study Year 2007. User report FOI-R–2348–SE, FOI, December 2007.

[8] Per Grahn, Peter Follo, Christina Grönvall, Gustav Haapalahti, Hans Habberstad, Mikael Karlsson, Fredrik Näsström, Johan Rasmusson, and Morgan Ulvklo. SEMARK Multipla sensorer för markläge - Slutrapport. Användarrapport FOI-R–2780–SE, FOI, Juni 2009.

[9] Andreas Hemmingsson. Composition of sensor services. Master's thesis, Royal Institute of Technology (KTH), School of Information and Communication Technology (ICT), 2009. TRITA.

[10] Hibernate ORM for Java. `https://www.hibernate.org/`. (senast besökt 03 november 2009).

[11] Jena - A Semantic Web Framework for Java. `http://jena.sourceforge.net/`. (senast besökt 10 november 2009).

[12] Pradeep Jonnaganti. Presentation of sensor data in opportunistic sensor networks. Master's thesis, Royal Institute of Technology (KTH), School of Information and Communication Technology (ICT), 2009. TRITA.

[13] A.F.M Sultanul Kabir. Graphical query builder to discover sensor information in opportunistic sensor networks. Master's thesis, Royal Institute of Technology (KTH), School of Information and Communication Technology (ICT), 2009. TRITA.

[14] F. Lantz, D. Andersson, R. Forsgren, E. Jungert, and B. Levin. Systemkoncept för övervakning av individstatus. User report FOI-R–2648–SE, FOI, December 2008.

[15] Yefeng Liu. Service composition workspace. Master's thesis, Chalmers, 2009. TRITA.

[16] Marianela García Lozano, Pontus Hörling, Farshad Moradi, and Edward Tjörnhammar. Supporting C2 with a Service Oriented Framework for Opportunistic Sensors and Sensor Networks. In *ICCRTS 2009*, June 2009. FOI-S–3224–SE.

[17] Carsten Magerkurth, Adrian David Cheok, Regan L. Mandryk, and Trond Nilsen. Pervasive games: bringing computer entertainment back to the real world. *Computers in Entertainment*, 3(3), July 2005.

[18] More On REST. `http://msdn.microsoft.com/en-us/magazine/dd942839.aspx`. (senast besökt 08 december 2009).

[19] Open Participation Software for Java. `http://wiki.ops4j.org/display/ops4j/Open+Participation+Software+for+Java`. (senast besökt 03 december 2009).

[20] OSGi Alliance. `http://www.osgi.org/`. (senast besökt 03 december 2009).

[21] OSGi Service Platform Release 4, version 4.2. `http://www.osgi.org/Main/HomePage`, September 2009. (senast besökt 10 november 2009).

[22] PAX - The OSGi station at OPS4J. `http://wiki.ops4j.org/display/ops4j/Pax`. (senast besökt 03 december 2009).

[23] RESTwiki. `http://rest.blueoxen.net/cgi-bin/wiki.pl`. (senast besökt 08 december 2009).

[24] SM4ALL-projektets Webbsida. `http://www.sm4all-project.eu/`. (senast besökt 10 november 2009).

[25] SOA & OSGi. `http://www.osgi.org/blog/2007/09/soa-osgi.html`. (senast besökt 03 december 2009).

[26] SOAP or REST. `http://stackoverflow.com/questions/76595/soap-or-rest`. (senast besökt 08 december 2009).

[27] SpringSource. http://www.springsource.org/. (senast besökt 03 december 2009).

[28] SpringSource dmserver. http://www.springsource.org/dmserver. (senast besökt 03 december 2009).

[29] Swedish Institute of Computer Science, forskning och design av Pervasive Games. http://iperg.sics.se. (senast besökt 16 november 2009).

[30] The Scala Programming Language. http://www.scala-lang.org/. (senast besökt 03 december 2009).

[31] US Defence Information Systems Agency (DISA) description of NECC. http://www.disa.mil/news/pressresources/factsheets/necc.html. (senast besökt 10 november 2009).

[32] Upkar Varshney. Pervasive Healthcare and Wireless Health Monitoring. *Mobile Networks and Applications*, 12(2-3), 2007.

[33] W3C Web Services Architecture. http://www.w3.org/TR/ws-arch. (senast besökt 10 november 2009).

[34] Wikipedia about Service-Oriented Architecture. http://en.wikipedia.org/wiki/Service-oriented_architecture. (senast besökt 10 november 2009).

[35] Wikipedia about Web Services. http://en.wikipedia.org/wiki/Web_service. (senast besökt 10 november 2009).

[36] WISENET Webbsida. http://www.wisenet.uu.se. (senast besökt 10 november 2009).

[37] Joel Wright, Christopher Gibson, Flavio Bergamaschi, Kelvin Marcus, Ryan Pressley, Gunjan Verma, and Gene Whipps. A Dynamic Infrastructure for Interconnecting Disparate ISR/ISTAR Assets (the ITA Sensor Fabric). In *Proc. 12th International Conference on Information Fusion, (FUSION 2009)*, pages 1393–1400, July 2009. Se även http://www.alphaworks.ibm.com/tech/fabric4sensors (senast besökt 16 november 2009).