

Utveckling av logik för handelsagent

Development of logic for a trade agent

Samuel Zayas
Marcus Johnsson

2005

EXAMENSARBETE
Informationsteknologi C
Nr: E3280IT



HÖGSKOLAN
Dalarna

EXAMENSARBETE, C-nivå i Informatik

| | | |
|---|--|-------------------|
| Program Informationsteknologi program, 140p | Reg nr E3280IT | Omfattning 10p |
| Namn Samuel Zayas Marcus Johnsson | Månad/År Juni 2005 | |
| | Handledare: Joakim Karlsson Examinator: Owen Eriksson | |
| Företag/Institution Experimenterande Svenska Radioamatörer | Handledare vid företaget Richard Niklasson | |
| Titel Utveckling av logik för handelsagent | | |
| Nyckelord agent, logik, handel, DSDM, handelslogik, handelsagent | | |

Sammanfattning

En agent är ett program som agerar för eller åt en individ med befogenhet från dem. Agenter kommunicerar med andra system, användare eller andra agenter beroende på den uppgift de är skapade för att lösa.

Swap Meet är en årlig loppmarknad för radiointresserade som sker under Radioträff Syd som arrangeras av vår uppdragsgivare Experimenterande Svenska Radioamatörer (ESR). ESR gav oss uppdraget att utforma en systemarkitektur för ett handelssystem som tillåter Swap Meet att bli mer än bara en årlig händelse.

Med hjälp av bl a Dynamic System Development Method (DSDM) har vi utformat en systemarkitektur för handel mellan agenter. Denna arkitektur är en grund för utveckling av ett sådant handelssystem.

Denna rapport syftar till att utveckla lämplig handelslogik för agenter och hur dessa agenter ska interagera med varandra.

För att utföra handel mellan ESR:s medlemmar på ett rättvist sätt har en av två teorier om genomförande av handel valts ut. Den utvalda teorin resulterade i ett auktionssystem som där agenter förhandlar genom budläggning.



DEGREE PROJECT in Information Systems

| | | |
|--|---|-------------------|
| Course Information technology program 140p | Reg number E3280IT | Extent 15 ects |
| Names Samuel Zayas Marcus Johnsson | Month/Year June 2005 | |
| | Supervisor: Joakim Karlsson Examiner: Owen Eriksson | |
| Company/Department Experimenterande Svenska Radioamatörer | Supervisor at the Company/Department Richard Niklasson | |
| Title Development of logic for a trade agent | | |
| Keywords agent, logic, trade, DSDM | | |

Summary

An agent is a program that acts for or in the place of another individual by authority from them. Agents communicate with other systems, users or other agents depending on the task they are created to solve.

Swap Meet is a yearly flea market for people interested in radio that takes place during the gathering "*Radioträff Syd*" that is held by our employer Experimenterande Svenska Radioamatörer (ESR). ESR gave us the task to create system architecture for a trade system that allows Swap Meet to be more than just a yearly event.

Using Dynamic System Development Method (DSDM) we designed system architecture with the purpose of trading between agents. This architecture is a foundation for development of such a trade system.

The purpose of this report is to develop suitable trade logic for agents and how these agents interact with each other.

To perform trade between the members of ESR in a just way one of two theories about trade negotiations has been selected. The chosen theory resulted in a auction system where agents negotiate through bidding.

Innehållsförteckning

| | |
|--|-----------|
| 1. INLEDNING | 1 |
| 1.1 BAKGRUND..... | 1 |
| 1.2 PROBLEMATISERING..... | 1 |
| 1.3 SYFTE..... | 1 |
| 1.4 MÅL..... | 1 |
| 1.5 FOKUS OCH AVGRÄNSNINGAR..... | 1 |
| 1.6 METODIK..... | 2 |
| 2 METOD | 3 |
| 2.1 LITTERATURSTUDIER..... | 3 |
| 2.2 KVALITATIV DATAINSAMLING..... | 3 |
| 2.3 DSDM - DYNAMIC SYSTEMS DEVELOPMENT METHOD..... | 4 |
| 2.4 DSDM:S UTVECKLINGSFASER..... | 4 |
| 2.4.1 Möjlighetsanalys..... | 5 |
| 2.4.2 Verksamhetsanalys..... | 5 |
| 2.4.3 Funktionell modelliteration..... | 5 |
| 2.4.4 Design- och utvecklingsiteration..... | 5 |
| 2.4.5 Implementering..... | 5 |
| 2.5 DSDM:S TIMEBOXING..... | 6 |
| 2.6 DSDM:S MOSCoW RULES..... | 7 |
| 3 TEORI & REFERENSRAM | 8 |
| 3.1 AGENTERS UTVECKLING..... | 8 |
| 3.2 ANVÄNDNINGSSOMRÅDEN..... | 9 |
| 3.3 FÖRSÄLJNINGSTEKNIKER..... | 9 |
| 3.4 AGENTENS DELSYSTEM..... | 10 |
| 3.5 SAMRÖRE MELLAN AGENTER..... | 11 |
| 3.6 KOMMUNICERANDE AGENTERS BETEENDE..... | 12 |
| 3.7 SYSTEMARKITEKTUR..... | 12 |
| 4 RESULTAT | 13 |
| 4.1 LITTERATURSTUDIER..... | 13 |
| 4.2 KVALITATIV DATAINSAMLING..... | 13 |
| 4.2.1 Funktionella krav..... | 13 |
| 4.2.2 Icke funktionella krav..... | 14 |
| 4.3 DSDM..... | 14 |
| 4.4 DSDM:S UTVECKLINGSFASER..... | 14 |
| 4.4.1 Möjlighetsanalys..... | 14 |
| 4.4.2 Verksamhetsanalys..... | 15 |
| 4.4.3 Funktionell modelliteration..... | 15 |
| 4.5 DSDM:S TIMEBOXING..... | 25 |
| 4.6 DSDM:S MOSCoW RULES..... | 26 |
| 5 ANALYS | 27 |
| 5.1 ANALYS AV AGENT..... | 27 |
| 5.2 HANDELSLOGIKEN..... | 28 |
| 5.3 DSDM..... | 29 |
| 6 SLUTSATSER OCH DISKUSSION | 30 |
| 6.1 AGENTEN OCH LOGIKEN..... | 30 |
| 6.2 METODVAL..... | 31 |
| 6.3 FÖRSLAG TILL VIDARE ARBETE..... | 31 |
| REFERENSER | 32 |
| BILAGOR | |

1. Inledning

Detta examensarbete behandlar agenter och logiken som används för att styra dessa. Enligt Alexander G. Moukas (1997) är agenter semi-intelligenta program som assisterar användaren i utförandet av tidskrävande uppgifter. Med ordet "handelsagent", som återkommer i rapporten, menar vi en agent som förhandlar om pris och kvantitet för inköp och försäljning av varor. I vårt arbete talar vi ofta om handelslogik vilket inte skall förväxlas med det som brukar kallas affärslogik i traditionell systemutveckling. I denna rapport innebär handelslogik vad det låter som, hur handel utförs.

1.1 Bakgrund

Experimenterande Svenska Radioamatörer (ESR) är en förening för alla som är intresserade av radioteknik och radiokommunikation. Föreningens målsättning är att verka för ökat tekniskt kunnande bland amatörradiointresserade.

ESR har ett antal aktiviteter tillgängliga för medlemmar så som träffar, seminarier och utbildningar. En av dessa träffar kallas för *Swap Meet*, en loppmarknad för radiointresserade, där medlemmarna kan köpa och sälja radiokomponenter av andra medlemmar.

1.2 Problematisering

ESR är en geografiskt spridd organisation och fysiska möten mellan medlemmarna försvåras på grund av detta. *Swap Meet*, som är en del av Radioträff Syd, arrangeras årligen på olika platser runt om i Sverige. Medlemmar från andra platser i landet finner det då svårt att ta sig till dessa träffar. Dessutom vill många medlemmar köpa och sälja komponenter under hela året vilket inte är möjligt i dagsläget. På något sätt vill medlemmarna ha möjlighet köpa och sälja sina komponenter och dessutom förhandla om pris och kvantitet. Kan man utveckla en agent för att utföra detta på distans? Hur skall handelslogiken till en sådan agent struktureras?

1.3 Syfte

Att utveckla en lämplig handelslogik för en agent som förhandlar med andra agenter angående köp och försäljning av utrustning.

1.4 Mål

Att utforma en lämplig systemarkitektur för en handelsagent.

1.5 Fokus och avgränsningar

Agenten kommer ej att vara självlärande utan kommer att använda sig av fördefinierad struktur och logik för sitt agerande.

1.6 Metodik

En grundläggande litteraturstudie för teknikmöjligheter genomförs tidigt, men kommer genomgående itereras då nya tekniker säkerligen kommer kräva ytterligare efterforskningar. Samtidigt kommer intervjuer och litteraturstudier ligga till grund för agentens funktionalitet. En systemutvecklingsmetod kallad *Dynamic Systems Development Method* (DSDM) kommer att användas för projektet som beskrivs mer noggrant i kapitel 2.

2 Metod

Då agenter är en relativt ny teknik krävs en omfattande litteraturstudie inom området. Samtidigt har en kvalitativ ansats tagits för att undersöka hur handelslogiken i agenten skall fungera. Eftersom ordförande i ESR bestämmer om systemet skall användas bör en genomgående intervju genomföras med just denna. Resultatet av intervjun blir då i princip en kravspecifikation från ESR. Genomförandet av de två ovanstående metoder är menat att ge en bild av dels hur agenter används idag och hur de är uppbyggda, dels hur agenten skall implementeras i ESRs verksamhet.

Nästa steg i arbetet blir därmed att designa en systemspecifikation som ligger till grund för systemutvecklingen. Detta kommer att ske med hjälp av en metod kallad DSDM, *Dynamic Systems Development Method*, (DSDM Konsortiet Sverige, 2005) som senare beskrivs mer noggrant i rapporten.

Då detta är klart kommer vi att presentera våra slutsatser och en diskussion kring dessa.



Figur 1 - Övergripande metodbeskrivning

2.1 Litteraturstudier

Litteraturstudier kommer i första hand att användas för generell kunskapsutveckling inom agentområdet och fördjupning inom metoden DSDM. Studier om hur en systemdesign bör utformas kommer att genomföras.

2.2 Kvalitativ datainsamling

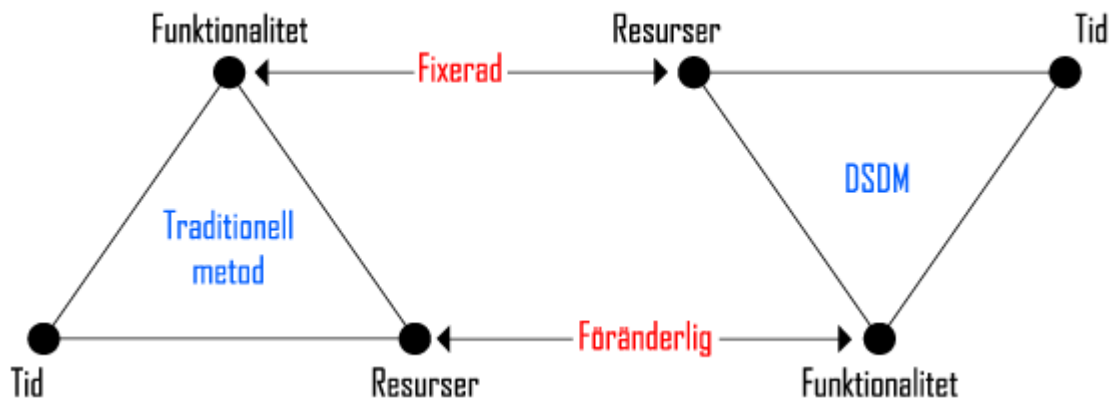
Enligt Holme & Solvang (1997) är validitet ett mått på om en fråga mäter eller beskriver det den avser att mäta eller beskriva. Detta innebär att man för det första måste ställa frågorna till rätt person, för det andra att ställa rätt frågor utifrån den kunskap respondenten innehar. För att hålla en så hög validitet på datainsamlingen som möjligt, kommer frågorna utformas på ett sådant sätt att respondenten inte leds in på olika spår. Ett problem som uppstår är att respondenten i vissa fall inte har tillräcklig kunskap om systemets begränsningar, då intervjuaren får förklara och riskerar att rikta frågorna. Holme & Solvang (1997) menar att det gäller att vara medveten om att försöka minimera den påverkan intervjuaren kan ha på en respondent. Enligt Häger (2001) är en intervju antingen strukturerad eller ostrukturerad. I detta arbete kommer vi använda en ostrukturerad intervju i två faser via e-post där första fasen innehåller frågor som låter respondenten svara relativt fritt och i andra fasen skickas vår tolkning av svaren till respondenten som får utvärdera dessa och besvara om tolkningen är korrekt.

2.3 DSDM - Dynamic Systems Development Method

"Genom ett användarorienterat angreppssätt med tydliga prioriteringar och absoluta deadlines levererar projektstyrningsmetoden DSDM dina lösningar på utlovad tid." (DSDM Konsortiet Sverige, 2005)

Enligt DSDM Konsortiet Sverige (2005) så är problemet med traditionella metoder för utveckling att krav lätt ändras under utvecklingstiden och då måste arbetet omvärderas. I dessa metoder är vanligtvis tid och resurser något som varierar beroende på hur långt man är från att uppfylla kravspecifikationen. Det betyder att beställare måste vänta på sin produkt till dess att kravspecifikationen helt uppfyllts. Ofta får då utvecklarna överskrida utsatt deadline och/eller anställa mer personal för att uppfylla kraven.

DSDM är en förlängning av RAD (*Rapid Application Development*) som fokuserar på projekt inom informationssystem där tid är en nyckelfaktor. Det är en iterativ och inkrementell metod vilket innebär att lösningen levereras i mindre delar som var för sig är användbara. Kravspecifikationen utformas av beställare i samråd med utvecklare som inför prioritering av kraven (kapitel 2.3.3 MoSCoW Rules). Tid och resurser är förutbestämt medan funktionaliteten är föränderlig (Figur 2). På detta sätt levereras affärsnytta redan tidigt i projektet eftersom delar av lösningen kan produktionssättas ibland månader innan hela lösningen är färdig. Exempelvis kan ett kundsystem levereras med endast de viktigaste funktionerna aktiverade först, så som att lägga till och spara kundinformation. Mindre prioriterade funktioner levereras allt eftersom projektet framskrider. Beställaren kan dock redan börja arbeta med sitt nya system innan all funktionalitet finns färdig.



Figur 2 – DSDM jämfört med traditionella utvecklingsmetoder

2.4 DSDM:s Utvecklingsfaser

Ett DSDM projekt innehåller fem faser som tillsammans bildar DSDM:s livscykel. Dessa faser är: Möjlighetsanalys, verksamhetsanalys, Funktionell modelliteration, Design- och utvecklingsiteration samt Implementering. Denna livscykel inleds med en Förstudie och avslutas med en Efterstudie. Förstudien handlar om att välja rätt projekt och efterstudien är direkt kopplad till underhåll av levererat system.

Förstudie, Möjlighetsanalys och Verksamhetsanalys görs sekventiellt och bestämmer spelreglerna för den iterativa och inkrementella utvecklingen. Hur de återstående faserna överlappar varandra beslutas av projektgruppen efter behov.

Eftersom målet med detta projekt är att presentera en systemarkitektur, och inte utveckla ett system, kommer faserna Design- och utvecklingsiteration samt Implementation ej att beröras. Samtliga faser finns dock beskrivna nedan.

2.4.1 Möjlighetsanalys

En viktig del av möjlighetsanalysen är att utvärdera i vilken omfattning DSDM kan användas för att komma fram till en lösning. Andra aspekter för denna fas är att definiera problemet som skall lösas, uppskatta pris samt om det är tekniskt möjligt att lösa problemet.

2.4.2 Verksamhetsanalys

Verksamhetsanalysen ska, som namnet angivet, identifiera verksamhetens behov. Resultatet av denna fas blir en definition av verksamhetsområde (*Business Area Definition*) som inte bara definierar verksamhetsprocessen utan också typer av användare som berörs av det tänkta systemet.

2.4.3 Funktionell modelliteration

Fokus för denna fas är att renodla de verksamhetsbaserade systemaspekterna, dvs att bygga överordnade funktionalitets- och informationskrav. Både Funktionell modelliteration och Design- och utvecklingsiteration består av cykler med fyra faser:

- Identifiera vad som ska utvecklas.
- Enas om hur och när detta ska utvecklas.
- Skapa produkten.
- Kontrollera att produkten stämmer överrens med förväntningar.

2.4.4 Design- och utvecklingsiteration

I denna fas ligger fokus på att finslipa systemet. Slutprodukten här blir ett färdigtestat system. DSDM har ingen specifik fas för testning eftersom detta görs fortlöpande under Funktionell modelliteration och Design- utvecklingsiteration genom de fyra cyklerna ovan (se Funktionell modelliteration). Testning är lika tidskrävande och viktigt i DSDM som i andra metoder, men det är fördelat över utvecklingsfaserna och sker iterativt.

2.4.5 Implementering

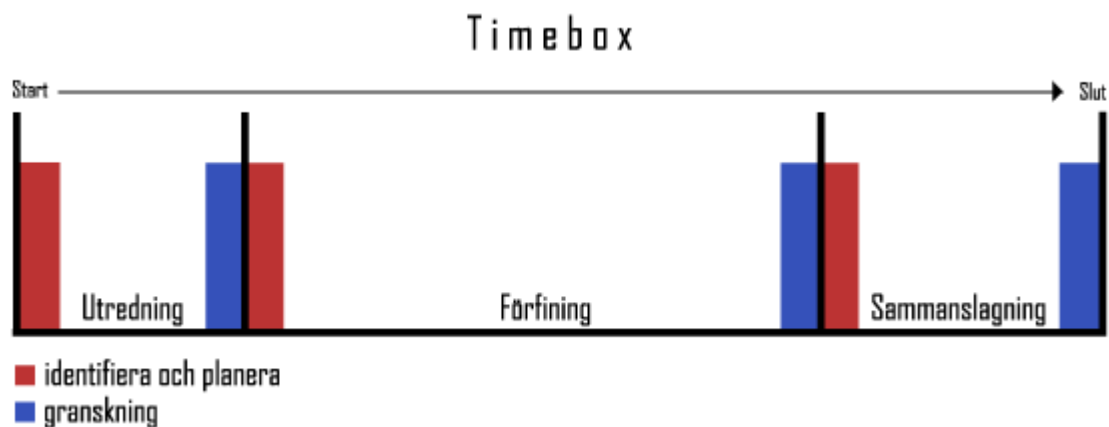
Här sker övergången från utvecklingsmiljö till funktionsduglig miljö. Detta innefattar att träna användare som inte varit med i projektgruppen.

Implementeringsfasen innehåller också en granskningsrapport för delleveranser, *Increment Review Document*, som är till för att summera vad projektet har uppnått i form av kortsiktiga mål. Dokumentet belyser alla krav som blivit identifierade under utvecklingen och systemets placering i förhållande till dessa krav. Eftersom DSDM är en inkrementell metod finns det fyra möjliga utfall från detta dokument:

- Allt är levererat och vidare utveckling är då ej nödvändigt.
- Ett nytt funktionellt område uppdagades under utveckling och således behövs en ny verksamhetsanalys.
- En mindre viktig (mindre prioriterad) del av funktionaliteten uteblev p g a tidspress. Utveckling återgår till Funktionell modelliteration och tillför denna funktionaliteten.
- Ett mindre tekniskt krav är inte tillfredställt och utvecklingen återgår till Design- och utvecklingsiteration för att bättra på detta.

2.5 DSDM:s Timeboxing

Inom DSDM används en metod som kallas för *Timeboxes* för att hantera flexibiliteten i ett projekt. Ett projekt delas in i flera *Timeboxes* med fasta start- och slutdatum.



Figur 3 - Faserna i en DSDM timebox

Varje *Timebox* innehåller tre faser:

- **Utredning**
En snabb utredning för att försäkra sig om att projektgruppen arbetar mot samma mål.
- **Förfining**
Att förfina och bygga utifrån kommentarerna från ovanstående utredning.
- **Sammanslagning**
Sista faser som knyter samman projektets olika delar.

Varje *Timebox* har ett fast slutdatum och på grund av detta måste vissa funktioner prioriteras framför andra. När det gäller prioritering använder man inom DSDM något som kallas för *MoSCoW Rules*.

2.6 DSDM:s MoSCoW Rules

Timeboxes är bestämda och slutgiltiga och således måste funktionaliteten variera beroende på hur mycket tid som finns tillgänglig. Grundläggande arbete måste göras klart först och mindre viktigt arbete stryks. För att förenkla denna prioritering används *MoSCoW Rules*:

M*ust* *haves* *fundamental* *to* *the* *projects* *success*.

S*hould* *haves* *important* *but* *the* *projects* *success* *does* *not* *rely* *on* *these*.

C*ould* *haves* *can* *easily* *be* *left* *out* *without* *impacting* *on* *the* *project*.

W*on't* *have* *this* *time* *round* *can* *be* *left* *out* *this* *time* *and* *done* *at* *a* *later* *date*.

(Märk att 'o' som förekommer i MoSCoW bara är där för att det ska vara möjligt att uttala ordet. Oftast skrivs dessa 'o' med gemener för att illustrera just detta)

Där finns alltså fyra prioriteringslägen: Måste finnas med, bör finnas med, kan utelämnas och utelämnas till ett senare datum. Dessa prioriteringar omprövas under projektets gång för att försäkra att de fortfarande är giltiga. Detta i sin tur försäkrar att projektet levereras i tid och att den viktigaste funktionaliteten kommer med.

3 Teori & Referensram

I detta kapitel kommer begreppet agent och dess kommunikation att förklaras mer noggrant. Av naturliga skäl är teorin om agenter inriktad mot kommunicerande agenter, eftersom förhandling inte kan genomföras med endast en part. Även de olika teorierna om handel som är intressanta för denna rapport tas upp.

3.1 Agenter utveckling

Idén om robotar som agerar helt eller delvis som människor är ingen nyhet. 1921 skrev Karel Capeks en pjäs om ämnet, R.U.R: *Rossum's Universal Robots*. På 1950-talet gjorde Isaac Asimov robotarna berömda genom att skriva boken "I, Robot". Ungefär samtidigt kom det första testet för att definiera om en dator sades tänka själv eller inte. Det kom att kallas *Turing Test of Artificial Intelligence*. Turing menade att en dator tänker själv om vi inte kan avgöra om vi kommunicerar med en dator eller människa. (David Braue, 2003)

Skillnaden mellan robot och agent är ibland hårfin. Dessutom definieras de olika av olika parter.

"En agent är ett system som försöker fullborda ett antal mål i en komplex, dynamisk miljö" (Pattie Maes, 1990) är en definition.

"Agenter är semi-intelligenta program som assisterar användaren i utförandet av tidskrävande uppgifter" (Alexander G. Moukas, 1997) är en annan.

Alexander G. Moukas (1997) ställer även upp tre viktiga delar om vad han anser att en agent måste vara eller kunna:

- De (agenterna) måste vara proaktiva, att utan direkta kommandon agera å användarens vägnar.
- De måste vara personliga, det vill säga de måste ta vara på användarens intresse och anpassa sig då de utvecklas över tid.
- De måste vara beständiga, antingen köras kontinuerligt eller spara dess tillstånd, så att användaren kan se agenten som en stabil entitet och utveckla en pålitlig relation med agenten.

Pattie Maes (1990) menar att robotar inte nödvändigtvis behöver omfatta någon form av intelligens, men agenter bör lära sig själva. Alexander G. Moukas (1997) menar att agenter skall kommunicera med andra agenter för att de skall kallas agenter. Även så kallade *spiders* som oftast är en form av datautsökare, kallas även agenter.

Man skiljer även på ensamma agenter och agenter som kommunicerar, kallade *single agents* och *multi agents*. Ensamma agenter handlar på egen hand och utför det arbete den är tillsagd att göra utan att få nya förutsättningar. De kommunicerande agenterna handlar autonomt men kan på grund av de agenter den kommunicerar med byta handlingsmönster. (Ibrahim F. Imam, Yves Kodratoff, 1997)

3.2 Användningsområden

Agenter används inom så många områden idag att det vore praktiskt taget omöjligt att rada upp dem. Vi använder oss av agenter i många fall utan att veta om det. Ett exempel kan vara sökmotorn Google (Google, 2005) som använder sig av agenter för att reda ut all information och lagra den på ett smart sätt. I Australien finns automatiserad gruvdrift, där maskiner får direktiv om hur de skall arbeta med hjälp av agenter. Agenterna kommunicerar här med varandra för att utföra arbetet på ett så effektivt sätt som möjligt. Inom flygvapnet används och forskas det kring agenter. The Soar Intelligent FORces project arbetar för att träna och delvis ersätta piloter med agenter med artificiell intelligens (The Soar IFOR Project, 2005).

3.3 Försäljningstekniker

”Handel, det överordnade begreppet för verksamheter som sysslar med omsättning av varor, tjänster och kapital mellan regioner och mellan människor. Handel är sålunda en överföring av nyttigheter från en säljare till en köpare utan att nyttigheten bearbetas. Handel baseras på avtal om t.ex. pris, kvalitet, mängd och leveranstid och innebär oftast att nyttigheter överförs från ett överskottsområde till ett underskottsområde. Handelns funktion är att på ett effektivt sätt sammanföra säljare och köpare på en marknad.” (Nationalencyklopedin 2005)

Två olika teorier gällande försäljning är intressanta för detta projekt. ”Den personliga försäljningen innebär personlig kontakt mellan säljare och kund, då båda två kan iaktta varandra och anpassa krav, erbjudanden och argument tills de kommer fram till en för bägge parter acceptabel lösning. Personlig försäljning kan ske på olika platser/---/” är en metod. (Nationalencyklopedin 2005)

Den andra teorin förklaras också bra av Nationalencyklopedin: ”auktion/.../, ett sätt att bjuda ut varor till försäljning. Auktionen leds av en auktionsförrättare. Den som lämnar det högsta budet, inropet, får rätt och skyldighet att köpa varan. Man skiljer mellan auktion dit allmänheten har tillträde, offentlig auktion, och privat auktion. Normalt sker auktion på så vis att auktionsförrättaren föreslår ett bud, varefter budgivning sker med överbud till dess att högsta bud har uppnåtts. Detta bud befästes sedan med klubbslag /---/” (Nationalencyklopedin 2005)

Båda teorier har egenskaper som kan tänkas passa in i en handelsagent logik.

3.4 Agentens delsystem

V.K. Murthy (2003) beskriver ett system med kommunicerande agenter genom att definiera ett antal undersystem. Bland annat hur dess miljö ser ut, hur den uppfattar saker från sin miljö och standardiserar detta. Ett annat undersystem är problemlösningssinnet, det vill säga vad agenten faktiskt arbetar med och har för lokala regler. Denna del omfattar ofta självinlärning och intelligens, även om det inte är något krav för att det skall kallas agent. V.K. Murthy (2003) beskriver agentens kunskap om sin organisation gällande kommunicerande agenter:

Varje agent måste kunna kommunicera med omvärlden eller andra agenter. Dess kunskap om organisation antas därmed innehålla all information om dess relationer till andra agenter. Som exempel, vilka agenter som kan kommunicera med varandra, databundenhet mellan olika agenter, motsättningar mellan olika agenter.

Ytterligare tre undersystem beskrivs som kallas *INTRAN*, *EXTRAN* och *EFFEKT*. *INTRAN* utför den interna uppdatering av ny information som kan tänkas komma från omvärlden. Om denna bearbetning av information sedan påverkar omvärlden eller inte är upp till dess system att avgöra. *EXTRAN* behandlar information och avgör om denna skall aktivera utförande av externa handlingar. *EFFEKT* behandlar hur agenten påverkat sin omgivning och lagrar denna information för att senare använda den vidare. Även denna del omfattar ofta självinlärning, genom att låta agenten hålla reda på vilka handlingar som påverkat sin omgivning och hur, för att kunna effektivisera detta genom att uppdatera sitt handlingsmönster i *INTRAN*.

Användningen av dessa system är beroende på vad agenten faktiskt skall utföra. Till exempel en agent som endast tar emot information och utför beräkningar som den sedan lagrar krävs i princip inte någon del av *EXTRAN* eftersom ingen information skall skickas.

Ett annat exempel kan vara en statisk agent som aldrig ändrar sin vy på omvärlden, utan endast matar ut information till andra och inte behöver något *INTRAN* undersystem.

3.5 Samröre mellan agenter

”Samröre är en interaktiv process mellan ett antal agenter som resulterar i varierande grad av samarbete, konkurrens och handlingar som leder till överenskommelser, samstämmighet eller motsättningar” (V.K. Murthy, 2003)

Samröre mellan agenter kräver att det finns någon form av protokoll för hur detta skall ske. Detta kan liknas vid att olika folkgrupper talar olika språk. I en digital miljö kräver detta att man inte har någon form av dialekter eller olika språk, då detta omedelbart kommer att feltolkas. Ett protokoll som utformats just för kommunikation mellan förhandlande agenter beskrivs av V.K. Murthy (2003) genom att agenten hanterar följande tillstånd eller handlingar:

1. Förslag: Agenten föreslår ett antal intentioner.
2. Godkännande: Förslaget godkänns för att utföra vidare handling.
3. Avslag: Förslaget avslås för att utföra vidare handling.
4. Modifikation: Detta förändrar någon del av det som föreslagits och föreslår detta nya förslag.
5. Inget förslag: Genererar inget samröre.
6. Avbryt: Avslutar samröre.
7. Rapportera överenskommelse: Här är förslaget accepterat och man utför vidare handlingar.
8. Rapportera misslyckande (överens om att man inte är överens): Avslutar samröre från alla inblandade parter.

Ett förhandlingsprotokoll bör omfatta två generella regler. Processen bör leda till ett begränsat antal handlingar och tillstånd. Den andra regeln är att man inte går in i cykliska mönster eller oändliga handlingar utan att man alltid hamnar i ett avslutat tillstånd.

Detta gäller förhandlingar i allmänhet och bör därmed gälla agenter. Man kan inte förhandla i evigheter utan att antingen komma överens eller avbryta förhandlingar.

3.6 Kommunicerande agenter beteende

Enligt V.K. Murthy (2003) finns 6 kännetecken som bestämmer en kommunicerande agents beteende:

1. Det finns en agent som påbörjar förhandling.
2. Varje agent kan vara aktiv eller inaktiv.
3. Alla agenter är inaktiva tills den startande agenten påbörjar handling.
4. En aktiv agent kan utföra lokala beräkningar, skicka och ta emot meddelanden och bli inaktiv.
5. En inaktiv agent blir aktiv endast om den mottager ett meddelande.
6. Varje agent kan bibehålla, uppdatera eller omvärdera sin kunskap beroende av nya meddelanden som mottages och processas. Om agenten utvärderar och uppdaterar sin kunskap meddelar denna de agenter som denna information berör.

Ett antal förutsättningar finns för att handel skall kunna påbörjas med hjälp av en agent, menar V.K. Murthy (2003). I agentens logik är detta initial data som används för att nå sitt slutgiltiga mål.

3.7 Systemarkitektur

För att kunna förmedla den information som förklarar hur ett system är uppbyggt, dvs systemarkitekturen, måste denna dokumenteras på ett sådant sätt att någon annan förstår vad som beskrivs. Detta är inte helt enkelt, men hjälpmedel finns i form av olika modelleringsverktyg. Vad och när man skall dokumentera tvistas ofta om. Vad denna dokumentation kallas är naturligtvis också beroende av vilken eller vilka metoder man använder. Östen Oskarsson kallar denna dokumentation konstruktionsdokument. Östen Oskarsson (1994) skriver: "Personligen tror jag att de här olika metodernas främsta uppgift är att förse en systemarkitekt med olika begrepp, angreppssätt och filosofier. I en verklig situation använder arkitekten allt han känner till och anser vara lämpligt. Att blint följa en viss metod ger inte en bra arkitektur. Det kan bara arkitektens yrkesskicklighet och intellekt åstadkomma".

UML (Unified Modeling Language) är i dag rådande världsstandard för att beskriva olika aspekter av informationssystem. UML används för att beskriva olika aspekter i ett system, ofta med hjälp av diagram som till exempel klassdiagram och sekvensdiagram.

4 Resultat

Under detta kapitel presenteras de resultat vi kommit fram till genom att använda de teorier och metoder som beskrivits i kapitel 2 och 3.

4.1 Litteraturstudier

I många metoder används UML för att beskriva systemdesign. Eftersom detta sätt att beskriva är allmänt accepterat har också vi valt att beskriva delar av detta system med hjälp av UML. I vår fas "Funktionell modelliteration" (kapitel 4.4.3 Funktionell Modelliteration) används UML för att beskriva klasserna i systemet och hur agenterna interagerar med hjälp av ett sekvensdiagram.

4.2 Kvalitativ datainsamling

Efter samtal med Richard Niklasson som i ESRs syfte arbetar med datasäkerhet och programmering blev det klart att den som kunde ge oss utförligast information om ESRs *Swapmeet* var ordförande Bengt Falkenberg, som även arbetar med teknik och drift inom ESR. Valet av respondent blev därför uppenbart, då Bengt har kunskap om ESRs organisation och vilka förutsättningar de har rent tekniskt.

Intervjuer med Bengt Falkenberg via e-post och telefon har resulterat i ett antal funktionella krav och ett antal ickefunktionella krav. Exempel på frågor presenteras i bilaga 1.

ESRs krav på systemet är enligt följande:

4.2.1 Funktionella krav

- Systemet skall hantera köp och försäljning mellan ESRs medlemmar.
- Användaren skall kunna välja om de skall sälja eller köpa en vara eller varor.
- Köp och försäljning skall omfatta pris och antal.
- Agenten ska innehålla lista på förval av komponenter så exakt matchning kan ske.
- Vid köp av en vara bestämmer användaren maxpris, antal och eventuellt om denna kan tänkas köpa färre än valt antal.
- Språket skall vara svenska.
- Användaren skall kunna kontrollera om agenten är färdig med sitt arbete eller inte.
- Användaren skall kunna begränsa tiden för hur länge en agent skall vara aktiv.
- Agenter skall kunna lösenordskyddas.

4.2.2 Icke funktionella krav

- Systemet skall kunna köras centralt hos ESR.
- Systemet skall vara plattformsoberoende.
- Agenterna skall följa FIPAs standarder.
- Logiken skall inte ge fördelar till någon av de förhandlande parterna.
- Kommunikation med systemet skall kunna ske över Internet.
- Systemet skall kunna hantera minst 10 agenter.
- En agent skall maximalt kunna vara aktiv i 4 veckor.

4.3 DSDM

DSDM fick anpassas en del för att passa in i vårt projekt då vi inte har som mål att leverera något system. Alltså togs faserna Design- och utvecklingsiteration samt Implementation bort. Dessutom användes alternativa metoder och teorier, så som UML och delar av Östen Oskarssons teorier om programutveckling, för att utveckla en systemarkitektur. Resultat från använda faser kan läsas om i kapitel 4.4 DSDM:s utvecklingsfaser.

4.4 DSDM:s utvecklingsfaser

För vårt projekt användes faserna Möjlighetsanalys, Verksamhetsanalys och Funktionell modelliteration. Resultatet från dessa är presenterade i följande tre underkapitel (4.4.1-4.4.3).

4.4.1 Möjlighetsanalys

Under denna fas beslutades det om DSDM var rätt metod att använda och i så fall i vilken utsträckning modellen kan hjälpa projektet - d v s vilka faser och verktyg som valts att använda.

Eftersom DSDM ger stor flexibilitet och är väldigt inriktad på att hålla tidsuppskattningar (*deadlines*) passade denna metod bra in på vårt projekt. Genom att använda *Timeboxes* och DSDM:s olika faser för projektutveckling kunde vi lättare planera vårt arbete.

Vi använde oss av Möjlighetsanalys (denna), Verksamhetsanalys och Funktionell Modelliteration för att nå fram till svaren på vår problematisering och uppnå syftet med rapporten. För rapportens mål, att presentera en systemdesign, kommer vi även ta användning av verktygen *timeboxing* och prioritering av krav utifrån *MoSCoW Rules*.

4.4.2 Verksamhetsanalys

En relativt begränsad verksamhetsanalys har utförts, eftersom det är ESRs medlemmar som privatpersoner som är intressenter för projektet snarare än organisationen. Kunskap om ESRs organisation är underordnande kunskap om hur dess *Swap meet* fungerar.

Varje år träffas fem radioklubbar i södra Sverige för att utbyta kunskap i bland annat form av föredrag, visa upp sina senaste alster och delta i olika projekt. Utöver detta finns en viss marknadshandel kallad *Swap meet*, där medlemmar kan sälja sina överblivna apparater och komponenter. Denna är utformad som en loppmarknad där varje enskild person har möjlighet att ställa ut sina varor för försäljning. Som exempel pågår *Swap meet* år 2005 under tre av de fyra dagar mötet pågår. (Experimenterande Svenska Radioamatörer, 2005)

Fördelar med att ha denna form av försäljning är att man kan titta och känna på varan, få reda på mer om dem och kanske hitta intressanta varor som man inte egentligen letat efter. Dessutom kan det vara enkelt att komma fram till pris då man ser skicket varan är i. Ett antal nackdelar uppstår genom att ha handel på detta sätt. I praktiken vill säkerligen alla medlemmar delta på de årliga mötena, men har inte möjlighet till detta, dessutom är det inte möjligt för alla medlemmar att få tid över till *Swap meet*, även om de är på mötet. Detta innebär att många medlemmar hamnar utanför möjligheten att komma över just den där komponenten som man letat efter.

Affärsnytta för ESRs medlemmar när det gäller *Swap meet* är inte nödvändigtvis detsamma som för dess övriga verksamhet där utbyte av kunskap och intressen är av högsta prioritet. Om utbyte av kunskap och intressen inom organisationen ens faller under kategorin affärsnytta kan diskuteras, dock görs detta inte i denna rapport.

När det gäller *Swap meet* ser man affärsnytta som att kunna dela med sig av sina varor till ett rimligt pris, det vill säga att båda parter får det de behöver. I ekonomiska termer beskrivs detta som ett över- och underskott från respektive part (Nationalencyklopedin 2005). Affärsnyttan är en balansering av detta över- och underskott utan några överdrivna marginaler.

4.4.3 Funktionell modelliteration

Enligt DSDM bör man i början av fasen *funktionell modelliteration* ställa upp de informations och funktionalitetskrav som gäller. Ingen speciell mall för dokumentering finns i DSDM för denna del, utan man använder sig av de kunskaper som finns tillgängligt för att utforma en sådan. Detta är inte detsamma som kravspecifikation från kund, utan är riktlinjer och specifikationer för hur systemet skall utvecklas internt.

Plattform och utvecklingspråk

Efter informella önskemål om plattformsoberoende i ett tidigt skede av projektet har JAVA valts som det programspråk som skall användas. Eftersom JAVA i många fall inom området är det språk som används, så passar detta projektet utmärkt.

Ett befintligt ramverk har valts ut för utveckling av agenten kallad JADE. Ramverket har utvecklats av Telecom Italia Lab och är *open source* mjukvara som faller under LGPL (Lesser GNU Public License). Det vill säga får användas och distribueras utan kostnad.

JADE följer FIPAs (Foundation for Intelligent Physical Agents) standarder för utveckling av agenter. Detta innebär att andra system som följer samma standarder kan kommunicera med system utvecklade med hjälp av JADE. Vi vet i nuläget inte om detta är en nödvändighet, men eftersom möjligheten finns är det rimligt att låta dörren vara öppen för detta.

Generella krav för agenten

Tanken bakom agenter i allmänna drag är att de får uppdrag som de sedan utför enligt bästa förmåga. Denna agent skall utföra två huvudsakliga mål, köpa, sälja varor. Agentens logik får dock ej vara sådan att den gynnar den ena parten mer än den andra, eftersom grundtanken är en rättvis handel mellan en organisations medlemmar, inte någon form av spekulationsmarknad. Varje agent skall antingen erbjuda en vara eller önska köpa en vara. Medlemmar har möjlighet att använda sig av obegränsat antal agenter, under förutsättning att värdsystemets utrustning kan hantera detta.

Initial data för agenten

Vi måste specificera vilka regler kommunicerande agenter kräver för att kunna genomföra en förhandling och ett eventuellt köp.

För en köpande agent måste därför följande information införas för att den skall kunna påbörja sitt arbete:

- Vad för vara eller varor som skall införskaffas.
- Ett maximalt pris för varan.
- Antal varor som önskas.
- Matcha exakt antal varor som önskas eller inte.

För en säljande agent behövs följande startregler:

- Vad för vara eller varor som skall säljas.
- Ett utropspris för varan.
- Ett utköpspris för varan.
- Eventuellt hur lång tid agenten skall vänta på bud.

Agentens beteende

Då vi följer V.K. Murthys teorier om de kännetecken en kommunicerande agent bör ha, måste varje ny agent i systemet meddela sin omgivning om att den existerar. Detta resulterar även i att man även följer övriga regler. Övriga agenter är inaktiva, om de inte är i processen att förhandla eller utför intern bearbetning, till dess att någon kontaktar dem.

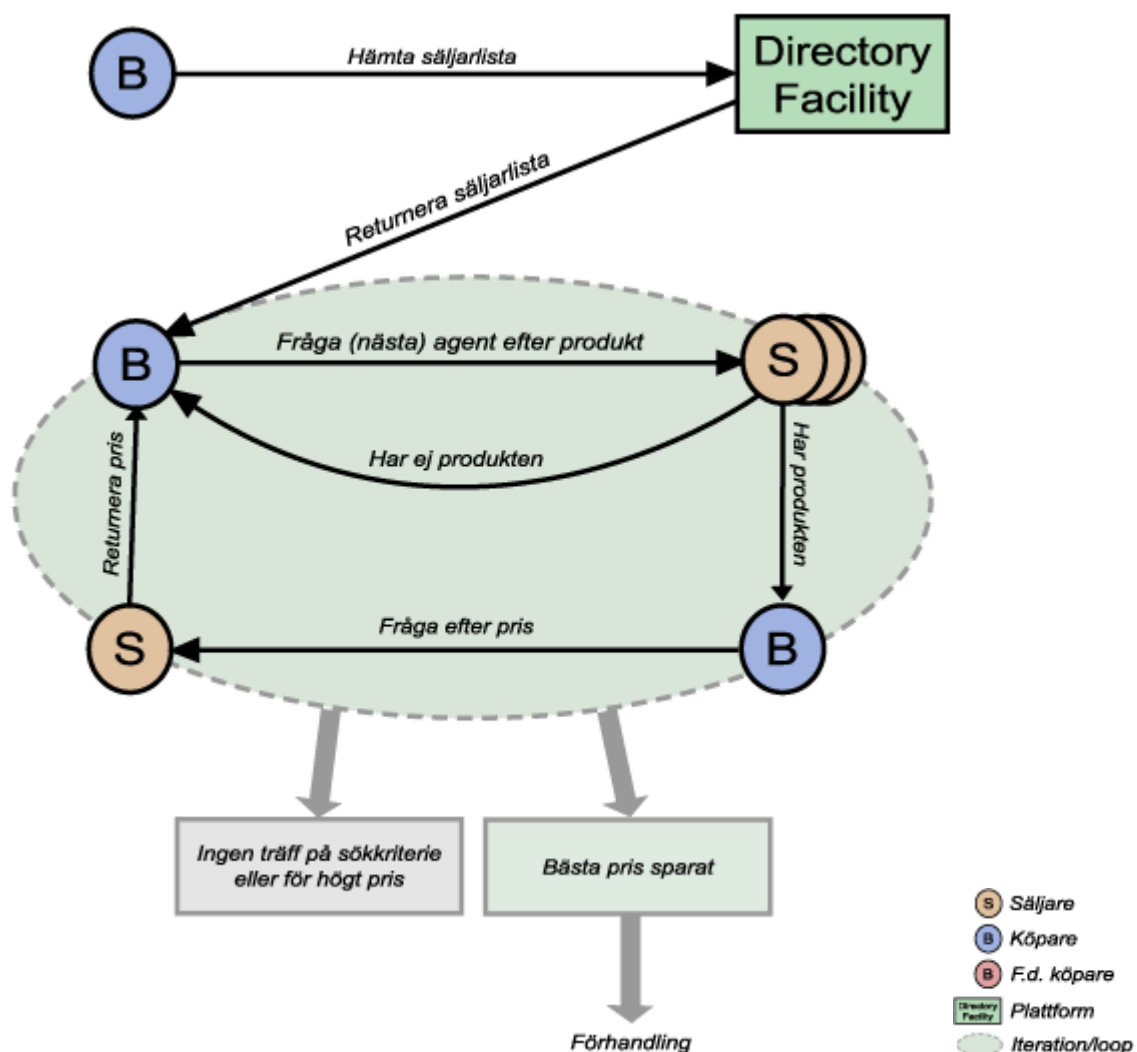
Vi har identifierat tre övergripande handlingar agenten kommer att utföra under sin livstid:

- Utsökning av motpart med rätt kriterier, såsom antal och pris.
- Budgivning och rapportering om budgivning.
- Avslut av försäljning när ett köp genomförts.

I den miljö agenten befinner sig i finns några olika scenarion som kan uppstå. Nedan kommer vi att beskriva hur respektive scenario utvecklar sig och hur agenten utför sina handlingar.

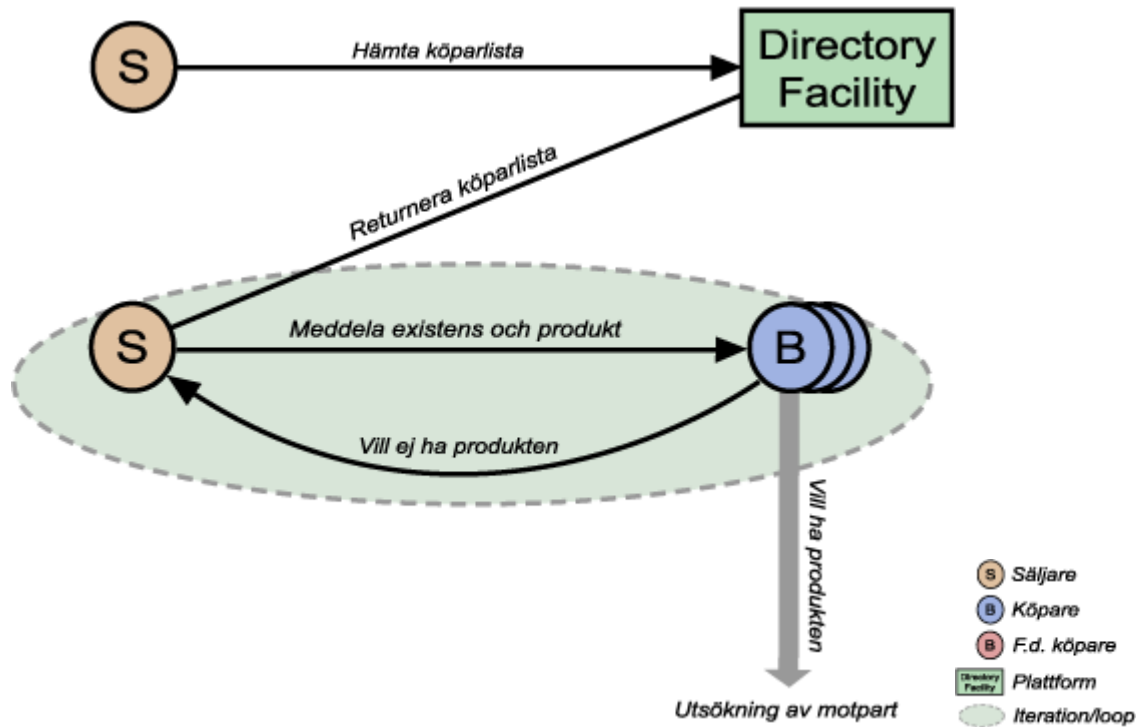
Utsökning av motpart

Då en agent som önskar köpa en vara träder in i systemet kommer denna att registrera sig med ett delsystem som fritt översätts Gula Sidorna (*Directory Facility*). Här registreras alla agenter som önskar att kommunicera med andra agenter i systemet. Efter registrering ber köparen om alla agenter som säljer varor utan urskiljning av vad de säljer. Köparen kommer sedan att gå igenom hela listan och fråga respektive säljare om vad de säljer och till vilket pris. Om säljaren inte har den produkt som önskas förkastas denna och nästa säljare kontaktas. Har säljaren det som önskas kommer priset att lagras och iterationen fortsätter. När alla säljare kontaktas kommer agenten, under förutsättning att den hittat någon som säljer rätt vara, att gå vidare till förhandling. Se figur 4 för logiskt schema av denna process.



Figur 4 - Utsökning av motpart initierat från köpagent

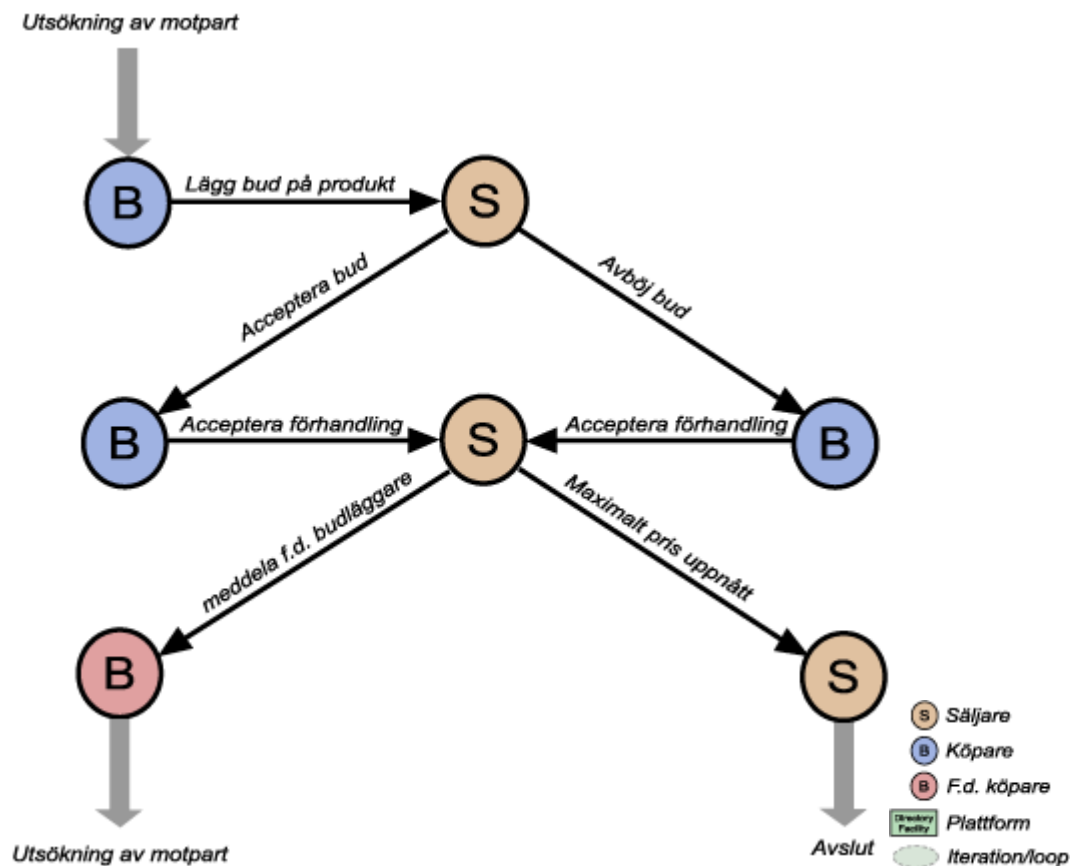
När en säljande agent träder in i systemet har denna en aning annorlunda beteende. Denna meddelar sin existens till *Directory Facility* och går sedan igenom köparlistan för att berätta om sin vara. Om någon köpare i denna lista skulle vara intresserad av vara kommer denna att utföra sin utsökning av motpart, eftersom den inte endast vill ta hänsyn till denna nya säljande agent, utan alla säljande agenter. Se figur 5 för logiskt schema av denna process.



Figur 5 - Utsökning av motpart initierat från säljagent

Förhandling

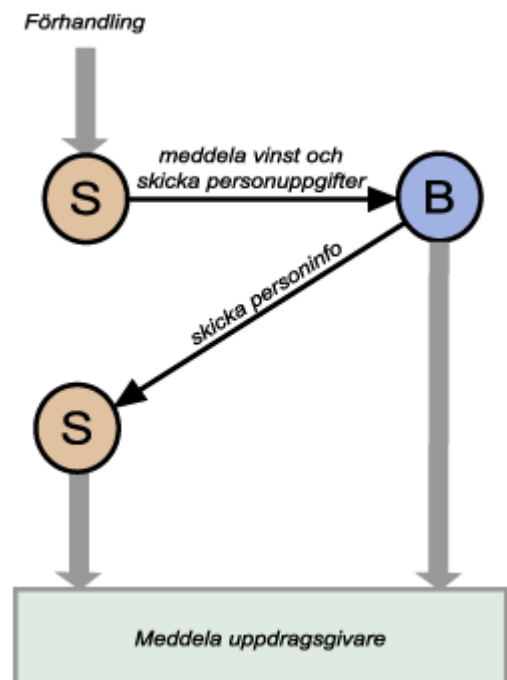
När en utsökning från köpande part är färdig har denna ett ID på den säljare som erbjuder det lägsta priset. Köpgenten kommer då att kontakta säljagenten och lägga ett bud på denna vara. Under vissa omständigheter kan säljaren tänkas neka budet, om tiden för auktionen skulle gå ut eller om den är i förhandlingsfas med en annan agent för tillfället. Annars kommer den säljande agenten att acceptera budet och meddela köparen detta. I detta fall kommer vidare handligas att utföras då den agent som tidigare var högsta budgivare måste meddelas att den blivit utbjuden. Om säljarens önskvärda pris (maxpris) uppnås kommer agenten gå till Avslut. Se figur 6 för logiskt schema över denna process.



Figur 6 - Förhandling mellan agenterna

Avslut

Efter en lyckad förhandling då säljaren nått önskvärt pris (maxpris) eller när en auktion nått sitt slutdatum kommer säljagenten meddela köpagenten om vinst samt skicka personuppgifter. Köpagenten skickar då i sin tur personuppgifter till säljaren varefter båda agenterna meddelar sina respektive ägare om resultat. Se figur 7 för logiskt schema över denna process.



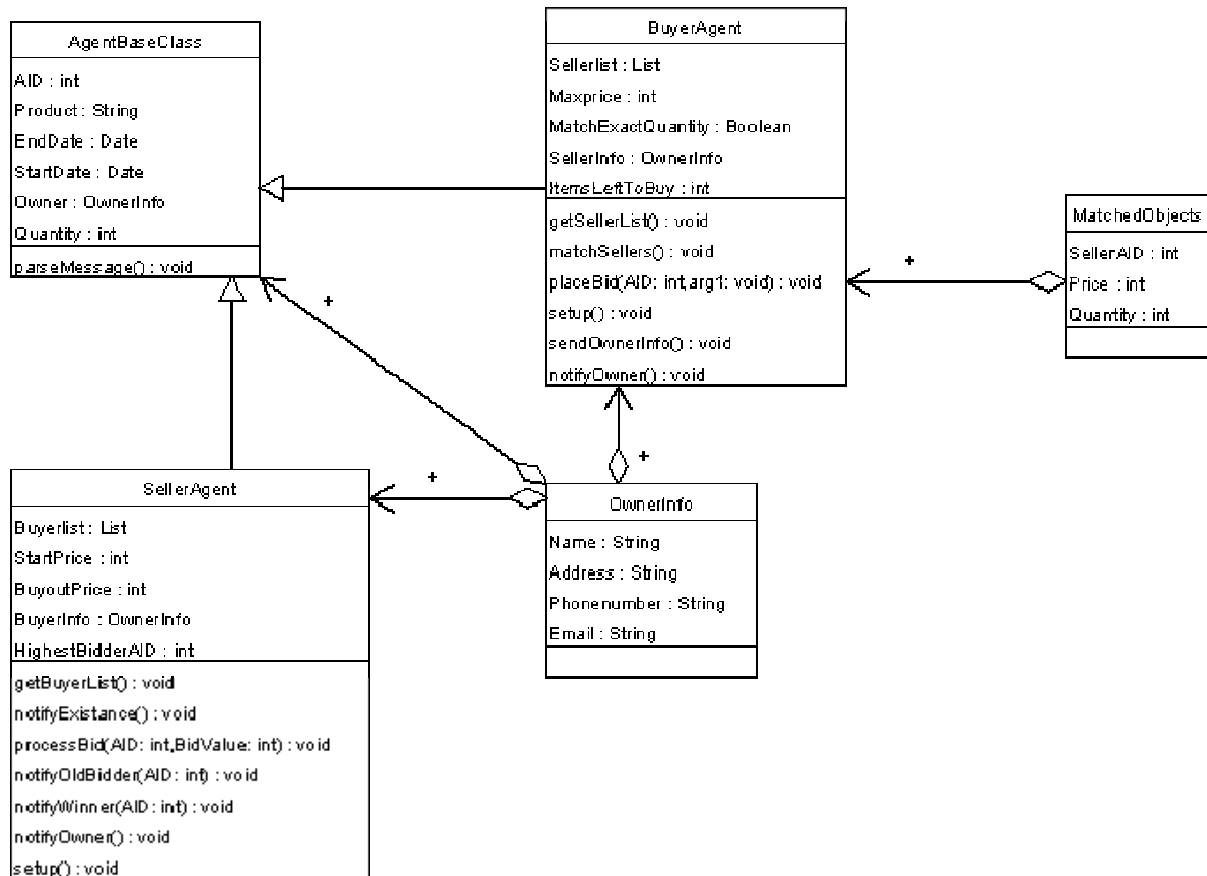
Figur 7 - Avslut mellan agenterna

Protokollanvändning

Vår användning av V.K. Murthys protokollteorier kommer begränsas något. I vissa fall krävs inte alla de steg som det fullständiga protokollet beskriver. Exempelvis gäller detta då en köpande agent gör en förfrågan till en säljande agent om den har varan finns eller inte. I detta fall kommer en förfrågan skickas och ett svar som beskriver antingen att ingen vara som matchar finns eller ett svar med antal och pris.

Klassdiagram

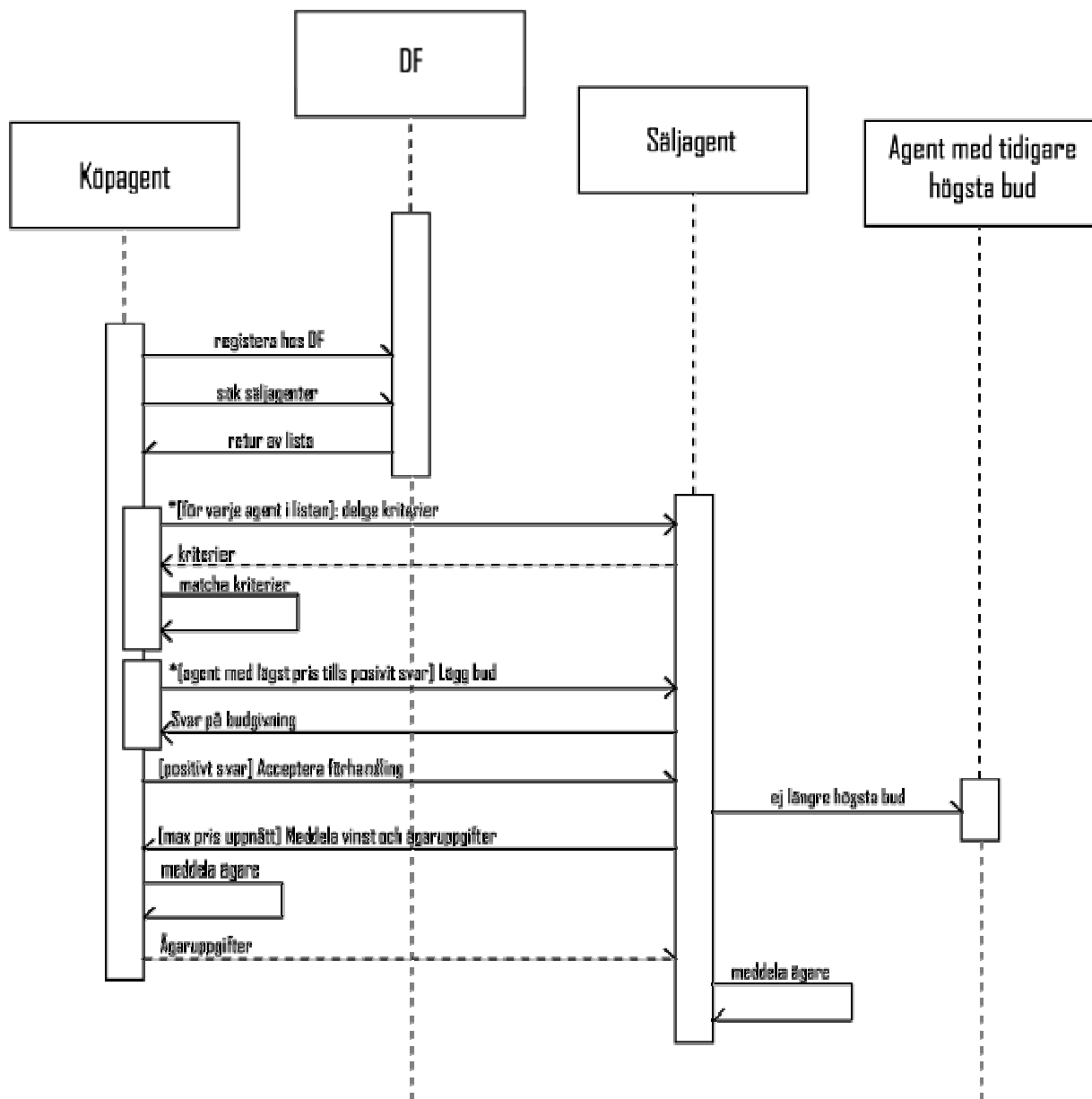
I vår interna design har vi kommit fram till ett antal klasser och strukturer som kommer krävas (Figur 8). En basklass omfattar alla de egenskaper respektive agent har gemensamt. Denna klass ärvs av de två typer av agenter som används i systemet utöver de systemagenter som används av ramverket. Två strukturer är utformade, en för ägarinformation som skickas mellan agenterna, en för förenkling när matchning av produkter och kvantitet utförs. Metoderna i respektive klasser utför den logik som beskrivits tidigare i kapitlet.



Figur 8 - Klassdiagram

Sekvensdiagram

Förklaring om hur kommunikation sker mer ingående mellan de olika agenterna i systemet presenterar vi i ett sekvensdiagram (Figur 9).

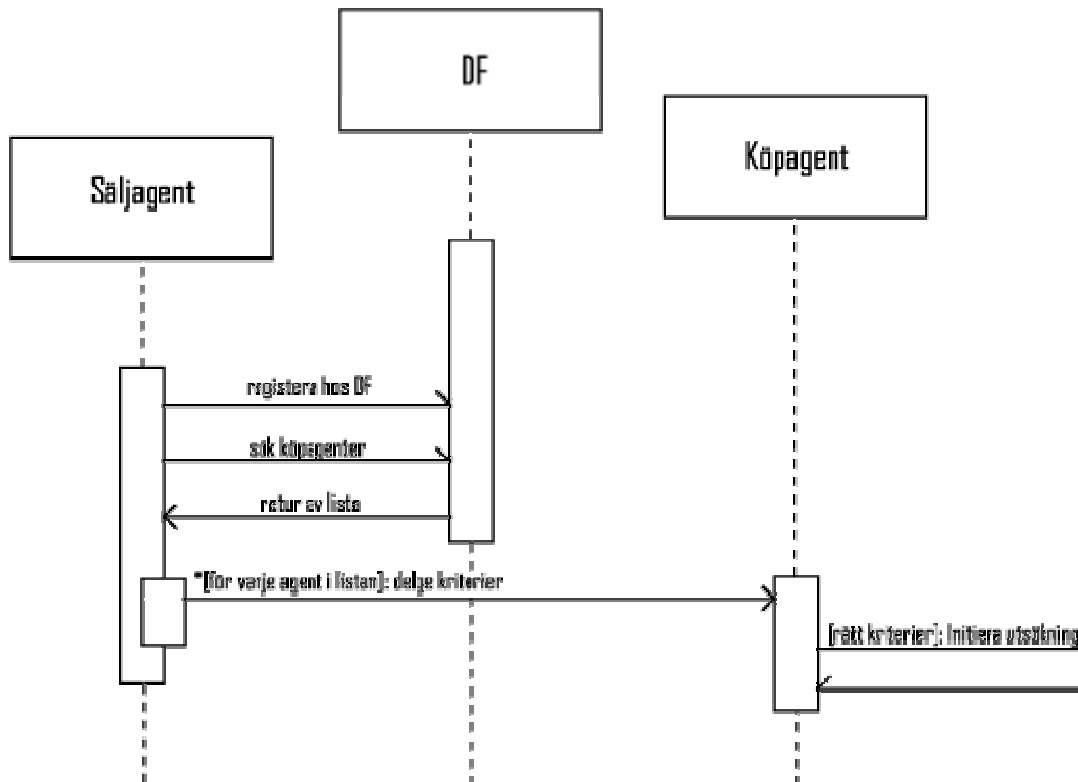


Figur 9 – Sekvensdiagram initierat av en köpagent

Detta diagram (Figur 9) är utifrån en köpagents perspektiv när den initieras i systemet. Köpagenten startar med att skicka två asynkrona meddelanden till DF (*Directory Facility*), registrering och en begäran om att få en lista med säljagenter. DF kommer därmed att returnera en lista, även denna skickas asynkront.

När köpagenten har mottagit listan från DF kommer denna iterera igenom listan och kontakta säljagenter och begära data om produkten säljagenter önskar sälja. Säljagenter returnerar detta varvid köpagenten matchar detta mot sina kriterier och lagrar i en lista. Detta kommer att ske synkront för att försäkra sig om att alla säljagenter får möjlighet att

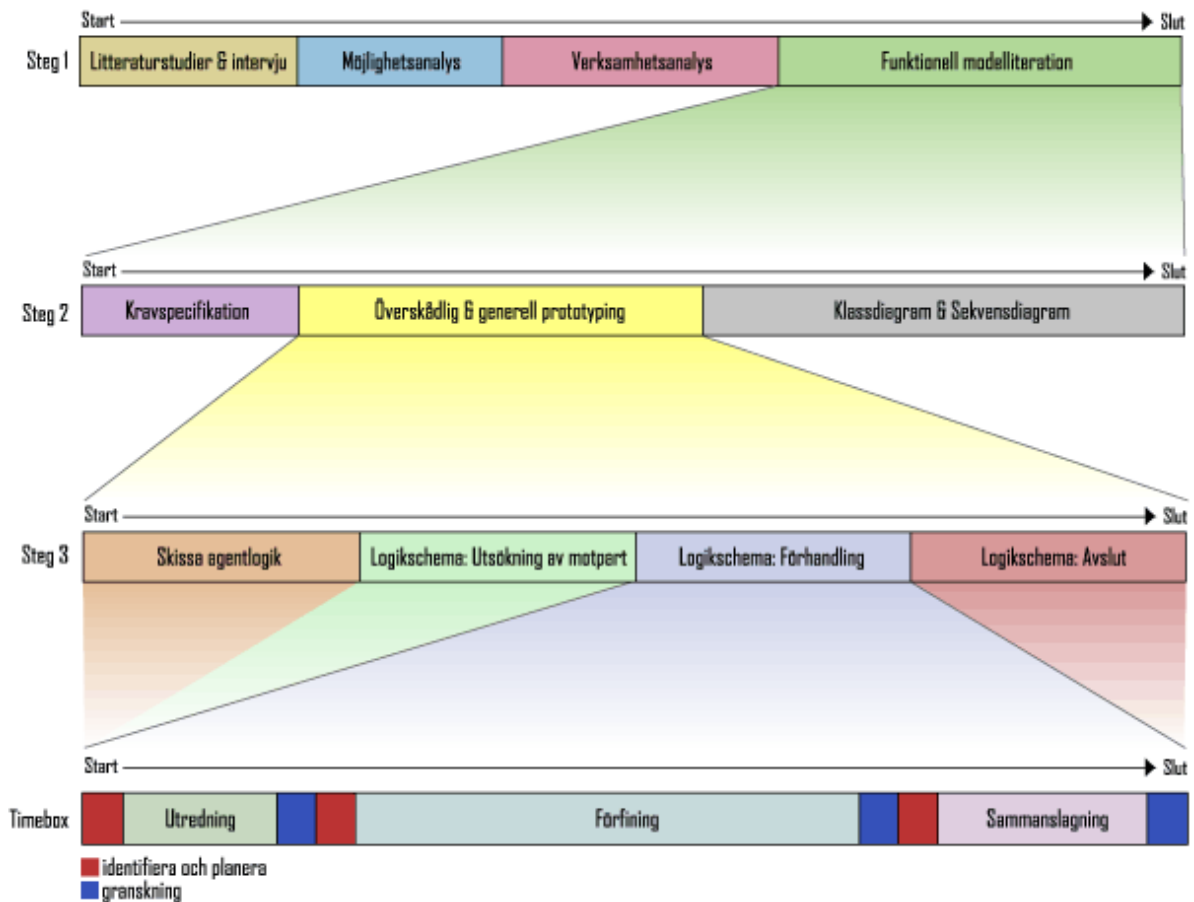
presentera sina kriterier. Därefter påbörjar köpagenten sin budläggning med start på lägsta pris. Även här kommer synkron kommunikation användas. Vid ett positivt svar, det vill säga att säljagenten accepterar budet skickar köpagenten en konfirmation om att budet accepterats för att senare försättas inaktiv tills den mottager ytterligare meddelanden. Om maximalt pris uppnås kommer säljagenten att meddela den köpagent som tidigare hade lagt det högsta budet om att detta inte längre gäller. Därefter meddelar denna den vinnande köpagenten samt skickar ägaruppgifter. Köpagenten meddelar i sin tur sin ägare och returnerar ägaruppgifter till säljagenten.



Figur 10 – Sekvensdiagram initierat av en säljagent

Ytterligare ett diagram (Figur 10) beskriver den interaktion en säljagent utför när den initieras i systemet. Den utför liksom köpagenten en registrering och en begäran av en lista av motparter, i säljagentens fall, köpagenter. Säljagenten itererar sedan igenom listan och meddelar varje köpagent sin information om dess produkt, varefter den går inaktiv. Köpagenterna matchar om dessa kriterier stämmer överens med vad den önskar. I sådant fall kommer denna att påbörja en ny utsökning enligt det tidigare diagram som beskrivits.

4.5 DSDM:s Timeboxing



Figur 11 - Timeboxing

Timeboxing gjordes för att kontrollera att vi mötte våra tidsuppskattningar i projektet. I bilden ovan (Figur 11) är de olika faser vi gått genom samt våra *timeboxes* åskådliggjorda.

Steg 1:

De första tre faserna i projektet – Litteraturstudier & Intervju, Möjlighetsanalys och Verksamhetsanalys – är direkt kopplade till varsin *timebox*. Funktionell modelliteration är uppdelad i tre faser som ligger i Steg 2 (Figur 11).

Steg 2:

Faserna i detta steg är: Kravspecifikation, Överskådlig & generell prototyping och Klassdiagram & sekvensdiagram. Kravspecifikation och Klassdiagram & sekvensdiagram är kopplade direkt mot två *timeboxes* medan Överskådlig & generell prototyping är uppdelad i ytterligare fyra faser.

Steg 3:

Detta steg innehåller fyra faser. Dessa är: Skissa agentlogik, Logikschema: Utsökning av motpart, Logikschema: Förhandling och Logikschema: Avslut. Var och en av dessa fyra faser slutar i en *timebox*.

Timebox:

Alla faser slutar i en detta steg. Dock kan *timeboxing* ske direkt i en fas, utan underfaser, så som sker med Litteraturstudie & Intervju. *Timeboxes* förklarades närmare i kapitel 2.5.

4.6 DSDM:s MoSCoW Rules

Med hjälp av MoSCoW Rules (kapitel 2.6 MoSCoW Rules), som är en del av vår systemarkitektur, har vi tagit fram en prioriteringslista av de funktionella och icke-funktionella krav som redovisats tidigare (kapitel 4.2 Kvalitativ datainsamling).

Prioritering sker med hjälp av en numerisk skala från 1 till 4 där 1 är högst prioritet och 4 lägst. Funktionella och icke-funktionella krav har prioriterats i samma lista då båda dessa är krav på hur systemet bör utformas.

Prioritet 1 - "Must have":

- Systemet skall hantera köp och försäljning mellan ESRs medlemmar.
- Användaren skall kunna välja om de skall sälja eller köpa en vara eller varor.
- Logiken skall inte ge fördelar till någon av de förhandlande parterna.
- Kommunikation med systemet skall kunna ske över Internet.
- Köp och försäljning skall omfatta pris och antal.

Prioritet 2 - "Should have":

- Vid köp av en vara bestämmer användaren maxpris, antal och eventuellt om denna kan tänkas köpa färre än valt antal.
- Språket skall vara svenska.
- Användaren skall kunna kontrollera om agenten är färdig med sitt arbete eller inte.
- Användaren skall kunna begränsa tiden för hur länge en agent skall vara aktiv.
- Systemet skall kunna köras centralt hos ESR.
- Systemet skall kunna hantera minst 10 agenter.

Prioritet 3 - "Could have":

- Systemet skall vara plattformsoberoende.
- Agenterna skall följa FIPAs standarder.
- En agent skall maximalt kunna vara aktiv i 4 veckor.

Prioritet 4 - "Won't have":

- Agenter skall kunna lösenordskyddas.

Denna prioritering är tänkt att användas som stöd för vidare utveckling och implementation av detta projekt.

5 Analys

I detta kapitel kommer den information som tagits fram i den teoretiska referensramen att sammanställas och sedan utvärderas utifrån de resultat vi presenterat.

5.1 *Analys av Agent*

Agenter används idag på många sätt för elektronisk handel, dock har vi inte funnit några tecken på att de i större utsträckning används i mindre system som det vi har utforskat. En av anledningarna är troligen att tekniken är ny och ses därför som komplicerad. Vi behagar att motsätta oss detta, agenttekniken är inte i sig komplicerad, den blir inte mer komplicerad än man gör den. Färdiga utvecklingsverktyg finns att använda som i många fall låter systemutvecklaren utforma logiken, men slipper problem med själva plattformen.

Något som vi måste ta hänsyn till när man utformar logiken är att varje agent är självstyrande och utför sin logik parallellt med andra agenter i systemet. Detta innebär att medan en agent utför sin beräkning på vilken säljare som har de bästa priserna, blir dessa produkter köpta av en annan agent. Agenterna i systemet fungerar på detta sätt, men vi anser inte att det bör uppstå någon orättvisa av två anledningar. För det första är de tider uträkningarna sker under mycket korta. För det andra är det den ägare (användare) som startar sin agent först som kommer att beräkna klart först och därmed lägga bud först.

Efter mycket vridande och vändande på J.K. Murthys teorier om hur agenter bör utformas, har vi inte funnit något som motsätter dessa. De teorier som J.K. Murthy beskriver faller in naturligt då man utvecklar systemet. Vissa delar av teorierna fanns redan med på skissnivån av vårt arbete, medan andra naturligt fogades in efter arbetets gång.

5.2 Handelslogiken

Två metoder för att utföra en rättvis handel har bedömts. Den första metoden som utvärderades var att man helt enkelt skulle komma överrens om ett pris, som baserats på säljarens minimipris och köparens maxpris. Detta såg vi först som en generellt bra metod, eftersom båda parter respektive köpt eller sålt till ett bättre pris än som förväntats. En enkel formel användes för att räkna på det pris som skulle avtals som såg ut enligt följande: $(\text{Köparens maxpris} - \text{säljarens minpris})/2 + \text{säljarens minpris}$. Detta pris skulle alltså hamna mitt emellan det pris båda önskar. Det uppstod dock flertalet problem med denna metod. Exempelvis:

En köpare bestämmer maxpris till 300kr och får varan för 250kr. Ganska snabbt bör köparen inse att säljaren säljer sin vara för 200kr $((300-200)/2+200 = 250)$. Köparen kommer därmed nästa gång lägga sitt maxpris till 200kr och får därmed varan till detta pris.

Ovanstående exempel visar att man kan utnyttja ett system som baseras på en "snäll" behandling. Detsamma gäller förstås för säljaren som kan öka sitt pris.

Ett auktionssystem som tar bud från agenterna kan förvisso göra att priserna blir högre, men ger mindre utrymme för utnyttjande av systemet. Säljaren kan visserligen ge bud på sina egna varor, men riskerar därmed att vinna auktionen. Om säljaren nu önskar att få bättre betalt för sin vara kan denna lika gärna sätta ett högre startpris.

Vi ansåg att risken för att någon medvetet försöker utnyttja systemet var relativt låg, men att ett auktionssystem troligen ligger närmare ett system som behandlar alla rättvist, eftersom man då lägger över hela ansvaret om priser på försäljare och köpare.

Hur man väljer sätt för att matcha rätt produkt kan bli ett stort problem. Det finns några valmöjligheter, men vilket sätt som bör användas är helt och hållet baserat på vem som skall använda agenten. Jämförs namnet på produkten exakt missar köparen produkter med mycket små variationer i namnet som än dock passar vad som är tänkt att köpas. Ett annat alternativ är att låta agenten rangordna hur nära produktens namn faktiskt passar. Passar namnet tillräckligt in, köper den helt enkelt produkter. Då är frågan, när är namnet tillräckligt passande? Kan vi lita på att agenten har uppfattat användarens önskemål eller behöver vi ta ett steg tillbaka och låta användaren avgöra om utvalda produkter passar? Vi anser att det återigen handlar om anpassning till vilket system man utvecklar agenten.

5.3 DSDM

DSDM är en relativt generell metod. Om man jämför DSDM andra traditionella utvecklingsmetoder så är dessa ofta mer detaljerade metoder – mer komplexa. Det innebär också att man har mindre flexibilitet när man använder sig av dessa traditionella utvecklingsmetoder.

DSDM innehåller fasta projektfaser och verktyg som *timeboxing* och *MoSCoW Rules* för att underlätta projektplanering, utveckling och leverans. Men det finns inga direkta krav på hur en fas skall genomföras. Här kan man antingen välja att ta hjälp från andra metoder, som innehåller starkare riktlinjer för de olika faserna, eller arbeta enligt eget tycke.

Med hjälp av DSDM:s *timeboxing* och prioritering genom *MoSCoW Rules* kan man känna sig säkra på att leverera sitt projekt i tid och att den viktigaste funktionaliteten finns med redan i tidigt i utvecklingsstadiet. Valen mellan utvecklingsmetoder var många när det gällde projekt likt detta. Skillnaderna är ofta inte stora mellan olika metoder men angreppssätt och detaljrikedom varierar.

Eftersom detta arbete inte slutar med en leveransbar produkt kunde vi inte fullborda DSDM:s livscykel. Faserna Design- och Utvecklingsiteration samt Implementering berördes alltså inte i vårt projekt.

DSDM valdes till planerings- och utvecklingsmetod för vårt arbete p g a just flexibiliteten.

6 Slutsatser och Diskussion

Här presenteras de slutsatser vi dragit utifrån vår analys. Dessutom förs en diskussion om alternativa val och vägar för utveckling.

6.1 Agenten och logiken

Forskningen inom agentområdet var mer omfattande än vad vi hade räknat med under början av vårt arbete. Informationen som relativt enkelt kan hittas i form av uppsatser, böcker och manualer är välarbetad och utförlig. När vi granskade dessa teorier upptäckte vi att man kunde få mycket hjälp av dem vid utvecklingen och att de stämmer väl överens med vårt eget tankesätt när det gäller agenter.

Utvecklingen inom området rent programmeringsmässigt för agenter är så pass långt gånget att det inte krävs några extrema färdigheter för att starta upp utvecklingsprojekt. Ett antal väl standardiserade ramverk för uppbyggnad av både agentens plattform och kommunikation existerar. JADE som vi haft under luppen igenom vårt undersökningsarbete är ett utmärkt sådant och rekommenderas vid agentutveckling.

Vi fokuserade på exakt matchning av produktnamn mellan agenter, då detta ställdes upp som ett gemensamt krav mellan oss och ESR. Andra alternativ för matchning kan vara att använda ontologi, d v s att agenten lär sig vilka ord som är besläktade med varandra och själv räknar ut om produktnamnen matchar, eller ett system för delvis matchning så som dagens sökmotorer på Internet fungerar. Problemet med det senare alternativet är att det kommer att finnas en viss felmarginal. Om detta inte är acceptabelt kan agenten återvända till användare med delvis matchningar så får denna sedan välja vilka som passar.

När det gäller hur mycket en agent faktiskt får bestämma själv krävs noggrann eftertanke. För att absolut lita på en agent måste denna besitta all kunskap som sin uppdragsgivare har, vilket troligtvis är omöjligt. Detta betyder att en viss felmarginal måste tas med i beräkningarna för vad en agent faktiskt utför. Förhållandet automatisering mot felmarginal måste övervägas, det vill säga, är nyttan med att låta agenten avgöra stor nog jämfört med risken att avgörandet blir felaktigt.

En fråga som dykt upp under arbetet är hur avancerad logiken kan och får bli? I detta system har vi ett antal startparametrar som agenten arbetar efter. Antalet och utformning av dessa står i direkt förhållande till hur avancerad logik agenten kan använda sig av. Idéer fanns i tidigare skeden om en förhandlingslogik som byggde på hur angelägen agentens uppdragsgivare var på att köpa eller sälja varan. Givetvis är detta ett intressant område, men i ESRs syften passar det föga eftersom vi inte vill gagna endera parten vilket är det primära syftet med en sådan logik, att tillskansa sig fördelar.

Det stod klart tidigt i vår undersökning att Auktionssystem var mest lämpat för ESR. Detta mycket på grund av att våra agenter måste utföra en rättvis förhandling för båda parterna. En köpagent får inte byggas "smartare" än en säljagent, och vise versa, då vi har hand om båda dessa parter.

6.2 Metodval

DSDM är en relativt generell metod och kan likas mer vid ett ramverk än en faktisk metod. Den innehåller utvecklingsfaser och verktyg för utveckling men saknar detaljerade riktlinjer för vad som skall uträttas och produceras från varje utvecklingsfas. I andra utvecklingsmetoder, så som XP (*Extreme Programming*) eller RUP (*Rational Unified Process*), finns dock mycket hjälp att hämta från dem som behöver starkare riktlinjer att följa. Detta är något som gör DSDM speciellt attraktivt; Att man kan ta hjälp från andra metoder för att fylla i hålen som DSDM lämnat efter sig. Frågan en projektgrupp bör ställa sig före de beslutar om utvecklingsmetod är hur pass styrda av riktlinjer och dokumentation de vill vara. Därefter kan DSDM användas i sin fullhet eller anpassas med hjälp av en annan utvecklingsmetod för att fylla de behov projektgruppen har.

Verktygen som DSDM tillhandahåller är till stor hjälp under hela projektet. De två huvudsakliga verktygen, *MoSCoW Rules* och *Timeboxing*, underlättar både planering och produktion.

6.3 Förslag till vidare arbete

Utveckla handelsagent med hjälp av DSDM

Att faktiskt utveckla denna agent med hjälp av DSDM är ett intressant projekt för att senare utvärdera metoden. DSDM är så pass omfattande och har såvitt vi funnit inte granskats särskilt genomgående i den akademiska världen. Vi frågar oss varför eftersom DSDM används inom många områden och företag som säkerligen önskar en utvärdering.

Fördjupning inom självlärande agenter

Detta skulle kunna utvecklas till mer än ett examensarbete i sig. Hur bör man utforma logik för en självlärande agent? Vad vill man att agenten ska lära sig, och varför? För en agent som handlar med varor uppstår ett stort behov av att kunna lita på agenten, vågar man låta agenten lära sig hur mycket som helst?

Mobila agenter och säkerhet

En teknik som idag finns är att kunna överföra kommunicerande agenter och dess programkod från en plattform till en annan. I ett slutet system, med flera plattformar för agenterna, är det möjligt att inga praktiska eller moraliska problem uppstår. Däremot om systemet är öppet och agenter från andra företag eller privatpersoner har rätt att köras på detta öppna system uppstår en hel del frågor. Vilka agenter skall tillåtas? Hur försäkras man sig om att ingen kan tillskansa sig privilegier som inte bör uppstå?

Referenser

Böcker

I. Holme & B. Solvang (1997) *Forskningsmetodik om kvalitativa och kvantitativa metoder. Studentlitteratur, Lund* (ISBN: 9144002114)

Ö. Oskarsson (1994) *Programutveckling i liten skala Studentlitteratur, Lund* (ISBN 91-44-46441-X)

Artiklar

I. F. Imam & Y. Kodratoff (1997) Intelligent adaptive agents: a highlight of the field and the AAAI-96 Workshop - Workshop on Intelligent Adaptive Agents, part of the 13th National Conference on Artificial Intelligence.
AI Magazine (ISSN 0738-4602)

P. Maes (1994) Modeling Adaptive Autonomous Agents
Artificial Life Journal, MIT Press, Vol. 1 & 2, No. 1-2. April 1994.

A. G. Moukas (1997) Amalthea: Information Filtering and Discovery Using A Multiagent Evolving System.
First issue of the Journal of Autonomous Agents and Multi-Agent Systems, 1998.

V.K. Murthy (2003) Interacting Agents in Networks for Negotiation and Decision Support
Proceedings of the 2003 Military, Government, and Aerospace Simulation Symposium.

Beynon-Davies & Williams (2003) The diffusion of information systems development methods.
Journal of Strategic Information Systems, Volume 12, Issue 1, Pages 29-46, 2003.

K. Southwell - Agile Process Improvement?
International TickIT (ISSN 1354-5884)

D. Braue (2003) - AI think, therefore I am
APC Magazine, <http://www.apcmagcom> december 2003.

Elektroniska källor

DSDM Konsortiet Sverige. <http://www.dsdm.se/>

Hämtad: 2005-05-02 14:32

Dynamic Project Management. <http://www.dpm.as/>

Hämtad: 2005-05-06 13:12

Wikipedia. <http://en.wikipedia.org/>

Sökord: DSDM, UML, agent, trade, timebox, MoSCoW Method.

Hämtad: 2005-05-08 10:14

Coley Consulting. <http://www.coleyconsulting.co.uk/>

Länk: Prioritisation (direktlänk: <http://www.coleyconsulting.co.uk/moscow.htm>)

Hämtad: 2005-05-09 08:15

Experimenterande Svenska Radioamatörer. <http://www.esr.se>

Hämtad: 2005-05-09 16:50

Nationalencyklopedins Internettjänst. <http://www.ne.se>

Sökord: Handel, auction, försäljning.

Hämtad: 2005-05-11 14:10

The Soar IFOR Project. <http://www.isi.edu/soar/>

Länk: IFOR Project (direktlänk: <http://www.isi.edu/soar/soar-ifor-project.html>)

Hämtad: 2005-05-13 07:00

Google Sverige. <http://www.google.se/>

Hämtad: 2005-05-20 23:59

Bilaga A - Exempel på frågor till respondenten

1. Beskriv generellt hur Swap Meet fungerar.
2. Vad anser du vara de viktigaste aspekterna för ett handelssystem åt ESRs medlemmar?
3. Hur många medlemmar har ESR som kan tänkas använda handelssystemet?
4. I hur stor utsträckning har ESR's medlemmar tillgång till egen dator och bredband?
5. Hur utförs handeln?
- 6a. Vad kan medlemmarna tänkas handla med för utrustning?
- 6b. Beskriv gärna vad olika delar och komponenter kan heta.
7. Önskemål om handelskriterier, hur ska kvantitet, tid hanteras?