

NIKLAS HALLBERG, JONAS HARALDSSON,
NINA LEWAU, JOACHIM HANSSON,
HELENA GRANLUND, THOMAS SUNDMARK, SUSANNA NILSSON

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.

Niklas Hallberg, Jonas Haraldsson, Nina Lewau,
Joachim Hansson, Helena Granlund, Thomas
Sundmark, Susanna Nilsson

Kravhantering: Best practice

Defects are not free. Somebody makes them, and gets paid for making them - Deming, E, 1986

Titel	Kravhantering: Best practice
Title	Requirements Engineering: Best Practice
Rapportnr/Report no	FOI-R--3264--SE
Rapporttyp /Report Type	Användarrapport
Månad/Month	September/September
Utgivningsår/Year	2011
Antal sidor/Pages	61 p
ISSN	ISSN 1650-1942
Kund/Customer	Försvarsmakten
Projektnr/Project no	E53334
Godkänd av/Approved by	Magnus Jändel
FOI, Totalförsvarets Forskningsinstitut	FOI, Swedish Defence Research Agency
Avdelningen för Informationssystem	Information Systems
Box 1165	Box 1165
581 11 Linköping	SE-581 11 Linköping

Sammanfattning

Förmågan att kostandeffektivt utveckla adekvata system är högst eftersträvansvärd. I praktiken är dock detta sällan fallet då många systemutvecklingsprojekt upplevs som mer eller mindre misslyckade. Orsaker till detta innefattar att inte lyckas leverera efterfrågade system samt svårigheter med att hålla tids- och budgetramar. Den aktivitet som pekats ut att ha störst påverkan på om systemutvecklingen ska nå ett lyckat resultat är kravhantering. Syftet med den studie som beskrivs i denna rapport är att beskriva state-of-the-art inom den vetenskapliga litteraturen i området Kravhantering, samt att relatera detta till Försvarens behov. Studien baseras på en systematisk litteraturstudie omfattande 375 artiklar publicerade under 2008-2011. Resultatet av studien presenteras i 16 övergripande kategorier som exempelvis Metoder och tekniker, Intressenter och deras krav, Prioritering, Kravdokument och språk, samt Modellering.

Slutsatsen är att det finns mycket att hämta för Försvarens behov inom området kravhantering, men att detta måste göras selektivt så att det som inhämtas fungerar i Försvarens nuvarande arbetsansats.

Nyckelord: Kravhantering, best practice, litteraturstudie

Summary

The ability to develop adequate systems in a cost-efficient way are essential. However, in practice this is rarely the case. Many systems development projects are perceived as more or less unsuccessful. Reasons for this include failing to deliver the requested systems and failing to keep the time and financial constraints. Requirements engineering has been identified as the activity having the greatest impact on systems development to achieve successful outcomes. The objective of the study, presented in this report, is to describe state-of-the-art of requirements engineering in the scientific literature and to relate this to the Swedish Armed Forces' needs. The study is based on a systematic literature review comprising 375 articles published during 2008-2011. Results of the study are presented in 16 categories, such as Methods and technologies, Stakeholders and their requirements, Prioritization, Requirements documentation and language, and Modeling

The conclusion is that there is a lot to obtain for the Swedish Armed Forces in the area of requirements engineering, but that this must be done selectively, so that the selected approaches fit the Swedish Armed Forces' current approach.

Keywords: Requirements engineering, best practice, literature survey

Innehållsförteckning

1	Inledning	8
2	Bakgrund	9
3	Metod	11
3.1.1	Litteratursökning.....	11
3.1.2	Litteraturanalys.....	13
4	Resultat	15
4.1	Ansatser	16
4.1.1	Goal-oriented requirements engineering	16
4.1.2	Aspect-oriented requirements engineering	18
4.1.3	Pattern-oriented baserad kravhantering	18
4.1.4	Problem frames.....	19
4.1.5	Software Product line engineering	19
4.1.6	Kravhantering inom agil systemutveckling.....	20
4.1.7	Användarfokus	21
4.2	Metoder och tekniker.....	21
4.2.1	Identifiera krav.....	21
4.2.2	Analysera krav.....	23
4.2.3	Säkerhetskrav	24
4.2.4	Tjänstbaserade system	24
4.2.5	Val av metod och teknik.....	25
4.3	Intressenter och deras krav.....	25
4.3.1	Användare- och verksamhetsbehov	26
4.4	Ontologi	27
4.5	Modellering.....	28
4.6	Kravdokument och språk.....	30
4.7	Spårbarhet.....	31
4.8	Prioritering av krav	32
4.9	Utvärdering av krav	33
4.10	Transformation	34
4.11	Releasehantering	34

4.12	Verktøgsstöd	35
4.13	Automatisering	36
4.14	Visualisering	36
4.15	Typer av krav	36
4.16	Tjänster	38
5	Diskussion	39
6	Referenser	42

1 Inledning

Inom Försvarsmakten läggs det ner omfattande resurser på att utveckla olika typer av ledningssystem, exempelvis SWECCIS och PRIO. Höga förväntningar finns på att dessa system effektivt ska stödja verksamheten på ett adekvat sätt. Svårigheterna med att konstandseffektivt utveckla system med rätt egenskaper är dock ett välkänt fenomen (Kasser, 2007). Det vill säga att lyckas med att ta fram rätt system, till rätt pris och i rätt tid, är en utmaning och i realiteten är andelen projekt som anses lyckade låg. Likaledes är nyttjandegraden låg hos de funktioner och finesser som implementeras i utvecklade system. Den enskilt högst bidragande aktiviteten till att lyckas med att utveckla system är en välgenomförd kravhantering (Konrad & Gall, 2008). Att genomföra kravhantering på ett tillfredställande sätt är dock långt ifrån en trivial aktivitet (dos Santos Soares, Vrancken & Verbraeck, 2011; Hallberg, Pilemalm & Westerdahl 2008).

Försummad kravhantering anses vara orsaken till de flesta och mest kostsamma bristerna i system. Kravhantering utgör ”bryggan” mellan verksamhetens behov och de realiseringar som utgör lösningen för dessa behov där en inadekvat kravhantering leder till system som varken uppnår uppsatta mål eller ger önskade effekter. Att tilldela tillräckligt med resurser till hanteringen av krav är därmed viktigt, då misstag och brister i de tidiga delarna av projektet blir kostsamma att rätta till senare (Boehm & Papaccio, 1998).

Syftet med studien som presenteras i denna rapport är att fastställa vad som är *state-of-the-art* och *best practice* inom kravhanteringsområdet vilket kan som underlag möjliggöra ytterligare förbättringar av kravhanteringen inom Försvarsmakten. Då syftet är att beskriva områdets *state-of-the-art* och ”*best practice*” medför det att frågeställningen är relativt öppen – Vad skrivs det för närvarande om avseende kravhantering inom den vetenskapliga litteraturen?

Denna rapport är den första från det FoT-finansierade projektet *Kvalitetsbaserad ledningssystemsutveckling*. Tillvägagångssättet i arbetet har varit en systematisk genomgång av den litteratur som publicerats om kravhantering under senare år, för att identifiera vad som skrivs inom området. Målet med projektet *Kvalitetsbaserad ledningssystemsutveckling* är att utveckla vetenskapligt kvalitetssäkrad och praktiskt förankrad kunskap, samt kompetens avseende kvalitetsbaserad ledningssystemsutveckling. Projektets medarbetare medverkar aktivt i olika projekt och initiativ inom Försvarsmaktens ledningssystemutveckling för att inhämta kunskap avseende svårigheter, kunskapsbehov och erfarenheter. Kunskapen som byggs upp kan sedan användas som grund för att genomföra forskningsverksamhet inom FoT-projektet, och därefter återföra kunskapen med avseende på möjliga lösningar till andra projekt och liknade verksamheter inom Försvarsmakten.

2 Bakgrund

Kravhantering (eng. requirements engineering) är ett omfattande kunskapsområde, inom vilket många ansatser, processer, metoder, notationer och tekniker utvecklats (Lauesen, 2002). Kravhanteringsområdet har inte lyckats att enas om en entydig och vedertagen definition av begreppet krav (eng. requirements). Till exempel definierar Siddiqi och Shekaran (1996) krav som *vad* ett system ska åstadkomma utan att beskriva *hur*, medan Sommerville och Sawyer (1997) menar att krav kan beskriva så väl problem, design som begränsningar i lösningar.

Traditionellt sett har kravhantering refererat till de tidiga delarna av systemutvecklingen, där krav på system identifieras, analyseras, specificeras och dokumenteras, innan aktiviteter som design, implementation samt verifiering och validering. De grundläggande aktiviteterna inom kravhantering är att: (1) identifiera verksamhetens och användares behov, (2) prioritera behov utifrån verksamhetens och användarnas perspektiv, (3) överföra identifierade behov till krav på systemet och att prioriteringar av behov överförs till prioriteringar på krav, (4) säkerställa spårbarhet mellan behov och krav, (5) förmedla krav på systemet till de som ska designa systemet samt (6) skapa underlag för validering och verifiering av krav (Hallberg, Timpka, & Eriksson, 1999).

Brister i kravhanteringen är den vanligaste orsaken till misslyckad systemutveckling (Konrad & Gall, 2008; Young, 2001). En orsak till svårigheterna med kravhantering är att varje utvecklingsprojekt är unikt. Varje enskilt fall föranleder ett analysbehov över vilka tillvägagångssätt som är bäst lämpade då projekt skiljer sig åt med avseende på utgångsläge, målsättning, tillgängliga resurser och kompetenser samt involverade intressenter.

Ett tidigt synsätt på kravhantering präglades av en produktfokusering. Systemets framgång mättes genom att avgöra till vilken grad systemet uppfyllde de krav som uttrycks i kravspecifikationen. Kravhantering sågs som en avgränsad aktivitet i utvecklingsprocessen vars enda koppling till övriga aktiviteter förmedlades via en kravspecifikation. Möjligheten att iterera krav uteblev till stor del och förutsättningarna saknades både för att validera krav och att verifiera utvecklade system mot krav. Den traditionella synen på kravhantering resulterade i färdiga system av otillräcklig kvalitet. Bristfällig återkoppling från användare och verksamhet under kravhanteringen ledde till att kraven på systemet inte svarade mot verksamhetens och användarnas reella behov (Kotonya & Sommerville, 1998). Produktfokuseringen ledde till en bristfällig koppling mellan behov och krav, samt att systemens nytta och användbarhet var långt ifrån optimal. En ensidig fokusering på produkten ledde dessutom ofta till otillräcklig dokumentation på grund av tendensen att blicka framåt mot produkten. Detta ledde till en negligering av själva systemutvecklingsprocessen som i sin tur ledde till problem med spårbarheten mellan behov, krav och slutligen design. Det

innebar även problem med systemkrav som var inkonsekventa eller ofullständiga (Kotonya & Sommerville 1998). Utebliven iteration, validering, verifiering och möjlighet till modifiering av krav resulterade i en oförändlig kravbild. Vid leverans innebar detta att det realiserade systemet inte motsvarade användarnas och verksamhetens behov och förväntningar.

En ansats sprungen från systemutvecklingsområdet för att undanhjälpa detta var utvecklingen av agil systemutveckling där både kravhantering och dokumentation dömdes ut. Ständig förändring var förutsättningen och via snabba iterationer skulle intressenternas önskemål uppfyllas successivt vilket inte heller visade sig helt oproblematiskt.

Nya ansatser uppstod även inom kravhanteringsområdet för att råda bot på skillnaden mellan vad intressenterna behövde och det som de utvecklade systemen tillhandahöll. Här kan nämnas ansatser för kravhantering så som *Problem frames* (Jackson, 2001) och *Goal-Oriented Requirements Engineering* (GORE) (Van Lamsweerde, 2001). Dessa ansatser utgår i högre grad från intressenternas ”problem” respektive ”mål”, för att åstadkomma system som bättre passade sin användning. Baserat på dessa ansatser har en stor mängd vetenskapliga bidrag skrivs. Vad som gäller i forskning inom kravhanteringsområdet idag beskrivs i avsnittet Resultat i denna rapport.

3 Metod

För att göra en grundlig och omfattande beskrivning av de begrepp och rön som diskuteras inom kravhanteringsområdet de senaste åren valdes en metodologisk ansats baserad på Systematic Literature Review (SLR). Målet med en SLR är att ge en väl underbyggd och kontrollerbar genomgång av tillgängligt vetenskapligt material kopplat till ett visst forskningsområde. Detta uppnås genom att följa en specifik och dokumenterad metod vid sökande efter och urval av material (Kitchenham, 2007). En tydlig redovisning av vilket material som ingår, samt varför, möjliggör granskning av oberoende part gällande hur utförligt arbetet skett. I den här studien har framförallt sökandet efter material skett genom SLR-metoden medan analysen av litteraturen utgjorts av en kvalitativ analys.

De övergripande aktiviteterna som genomförts är: (1) Litteratursökning och (2) Litteraturanalys. Nedan följer en mer utförlig beskrivning av de genomförda aktiviteterna.

3.1.1 Litteratursökning

Ett antal tidskrifter, magasin och konferenser relevanta för kravhanteringsområdet identifierades inledningsvis (Tabell 1) och mot dessa inriktades litteratursökningen. Genomsökningen genomfördes under juni 2011 och för att fånga aktuell forskning avgränsades urvalet till material publicerat mellan 2008 och 2011. En tidskrift och tre konferenser ansågs högrelevanta och från dessa inkluderades samtliga publiceringar inom tidsspannet.

I Tabell 1 redovisas de källor som använts och genom vilken sökmotor eller indexeringstjänst genomsökningen skedde. I de fall genomsökningen inte resulterade i några träffar markeras det i åtgärds-kolumnen med *Inga resultat*. De källor som ansågs högrelevanta markeras i samma kolumn som *Samtliga inkluderat*. För övriga källor som markerats med *Genomsökt* innebär det att de resulterat i minst en sökträff, vilken inkluderats i dataanalysen.

Genomsökningen baserades på en ihopsatt söksträng som inkluderade olika engelska benämningar på kravhantering. Den generiska söksträng som användes, med vissa modifieringar beroende på sökmotorns krav på inmatning, var:

"requirements engineering" or "requirements analysis" or
 "requirements specification" or "requirements acquisition" or
 "requirements elicitation" or "needs engineering" or "needs
 analysis" or "needs specification" or "needs acquisition" or "needs
 elicitation"

Tabell 1 Genomsökta källor

Namn på källa	Typ av källa	Sökmotor/ indexeringstjänst	Åtgärd
Human Factors	Tidskrift	SAGE Journals online	Inga resultat
Human-Computer Interaction	Tidskrift	Taylor & Francis Online	Inga resultat
IEEE International Requirements Engineering Conference	Konferens	IEEEExplore	Samtliga inkluderat
IEEE Software	Magasin	IEEE Xplore	Genomsökt
IEEE System Journal	Journal	IEEE Xplore	Inga resultat
Information & Management	Tidskrift	ScienceDirect	Inga resultat
Information and software technology	Tidskrift	ScienceDirect	Genomsökt
Interacting with Computers	Tidskrift	ScienceDirect	Genomsökt
International Conference on Engineering of Complex Computer Systems (ICECCS)	Konferens	IEEE Xplore	Genomsökt
International Workshop on Requirements Engineering Visualization (REV) 08-10	Konferens	IEEE Xplore	Samtliga inkluderat
Journal of Defense Modeling & Simulation Applications, Methodology, Technology	Tidskrift	SAGE Journals Online	Genomsökt
Journal of Systems and Software	Tidskrift	ScienceDirect	Genomsökt
Journal of Systems Architecture	Tidskrift	ScienceDirect	Inga resultat
Journal of systems science and Systems Engineering	Tidskrift	SpringerLink	Genomsökt
Requirements engineering	Tidskrift	SpringerLink	Samtliga inkluderat
Scandinavian Journal of Information Systems	Tidskrift	SJIS Archive	Inga resultat
Software and Systems Modelling	Tidskrift	SpringerLink	Genomsökt
Workshop on Requirements Engineering (WER) 08-11	Konferens	WERpapers	Samtliga inkluderat

I samtliga fall begränsades genomsökningen till metadata om artiklarna, så som titel, sammanfattning och nyckelord. Genomsökningen baserades alltså inte på artiklarnas hela text. Allt material som genomsökningen resulterade i sparades ner för analys, vilket totalt var 375 artiklar. Med *artikel* avses tidskriftartiklar, magasinartiklar, konferensbidrag och workshopbidrag.

3.1.2 Litteraturanalys

De 375 artiklarna granskades primärt utifrån titel, sammanfattning och slutsatser. Granskningen innebar alltså inte att hela artikeln analyserades. Detta beslut grundade sig på antagandet att artikelns viktigaste bidrag nämns antingen i sammanfattningen eller i slutsatserna. Granskningen delades upp mellan sju personer som var för sig granskade mellan 20 och 70 artiklar. Varje artikel bedömdes utifrån 23 olika områden, vilka formulerats efter en preliminär genomgång av materialet. Vid granskningen bedömdes om artiklarna berörde något eller några av följande intresseområden:

- Ontologi
- Produktnära / Utvärdera system mot krav
- Hantera krav
- Typer av krav
- Prioritering av krav
- Välja metod
- Användare och verksamhetskrav
- Intressenter
- Utvärdera krav
- Kravdokument
- Tjänster
- Domain engineering / Software Product line engineering
- Verktögsstöd
- Transformation
- Automatisering
- Visualisering
- Spårbarhet
- Ansats
- Metod
- Teknik
- Modellering
- RE och andra metoder
- Övrigt

Varje granskare noterade för varje artikel om artikeln berörde något av de bestämda intresseområdena. Efter genomgången användes dessa noteringar för att filtrera fram de artiklar som var knutna till var och ett av intresseområdena. Ett eller flera intresseområden tilldelades sedan olika personer som gjorde en sammanfattning för varje intresseområde.

Efter denna sammanfattning analyserades åter uppdelningen av intresseområden där ett par områden slogs samman för att resultera i den slutliga indelningen:

- Ansatser
- Metoder och tekniker
- Intressenter och deras krav
- Ontologi
- Modellering
- Kravdokument och språk
- Spårbarhet
- Prioritering av krav
- Utvärdering av krav
- Transformation
- Releasehantering
- Verktögsstöd
- Automatisering
- Visualisering
- Typer av krav
- Tjänster

4 Resultat

Kravhantering är enligt Konrad och Gall (2008) utan tvekan den viktigaste aktiviteten när komplexa, mjukvaru-intensiva system ska utvecklas. Ingen annan aktivitet har större inverkan på framgången i ett systemutvecklingsprojekt. Kravanalytikern (eng. requirements engineer) utgör en brygga mellan personer som normalt inte samverkar, mellan verksamhet (eng. business) och teknik, samt mellan problem och lösning (Sim, Alspaugh, & Ban Al-Ani, 2008). För kravanalytikern innebär det att mjuka kompetenser så som kommunikationsförmåga, diplomati, förhandlingskapacitet och ledarskap kan vara avgörande för att nå framgång (Sim, Alspaugh, & Ban Al-Ani, 2008; Dieste, Juristo, & Shull, 2008; Fricker & Glinz, 2010; Konrad & Gall, 2008).

Att lyckas med identifiera krav anses vara svårt, samt svårigheten ökar i takt med systemens komplexitet (Konrad & Gall, 2008; Soares, Vrancken, & Verbraeck, 2011; Rosado, Fernández-Medina, López, & Piattini, 2010), mängden användare (Jantunen, Gause, & Wessman, 2010) och antalet samverkande system. Ytterligare svårigheter orsakas av oklara och otydligt uttryckta projektmål och önskade funktioner (Alexander, 2011). Beslut tas ofta om systemets funktionalitet baserat på otillräcklig kunskap avseende kravbild, vilket medför stor risk att systemet senare i utvecklingsprocessen visar sig ha bristfälliga egenskaper (Feather, Cornford, Hicks, Kiper, & Menzies, 2008). Forskning som sker inom området kravidentifiering (eng. requirements elicitation) pekar på vikten av att ha en adekvat sammanställning av de ”verkliga” kraven – krav baserade på intressenternas behov.

Användare är en intressentgrupp som måste beaktas vid identifiering av krav. Ofta saknar användare kunskap i både kravhantering och systemutveckling, och kan i vissa situationer behöva stöd i kommunikationen med kravanalytiker och utvecklare. Detta kan ges i visuell form med stöd av prototyper (Pérez & Valderas, 2009).

Inom agil systemutveckling ses förändring som en grundförutsättning, och förhåller sig till detta på ett systematiskt sätt. Det medför att agil kravhantering väsentligt skiljer sig från traditionell kravhantering. Exempelvis ses identifieringen av krav som en aktivitet som utförs under hela utvecklingsprocessen, inte enbart i inledningen av den (Racheva, Daneva, Sikkil, Wieringa, & Herrmann, 2010; Cao & Ramesh, 2008).

Nedanstående avsnitt beskriver resultatet av litteraturstudien i kategorierna (1) Ansatser, (2) Metoder och tekniker, (3) Intressenter och deras krav, (4) Ontologi, (5) Modellering, (6) Kravdokument och språk, (7) Spårbarhet, (8) Prioritering av krav, (9) Utvärdering av krav, (10) Transformation, (11) Releasehantering, (12) Verktögsstöd, (13) Automatisering, (14) Visualisering, (15) Typer av krav och (16) Tjänster.

4.1 Ansatser

Det finns en flora av olika ansatser för att genomföra kravhantering. Den, i litteraturstudien, mest omtalade är *Goal-Oriented Requirements Engineering* (GORE), vilken ligger till grund för en betydande del av den forskning som bedrivs inom kravhantering idag. Förutom GORE finns det ett flertal andra orienteringar vilka enligt Alexander (2011) inkluderar *Stakeholder-oriented* (SORE), *Context-oriented* (CORE), *Scenario-oriented* (ScORE) och *Priorities-oriented* (PORE). Ytterligare ansatser är *Aspect-Oriented* (AORE) (Oliveira, Araújo, & Amaral, 2010; Rashid, 2008; Weston, Chitchyan, & Rashid, 2009; Weston, Chitchyan, & Rashid, 2008), *Task-oriented* (TORE) (Adam, Doerr, Eisenbarth, & Gross, 2009), *Value-oriented* (Glinz, 2008), *Communication-oriented* (González, Ruiz, España, & Pastor, 2008), *Feature-Oriented* (Waldmann & Jones, 2009) och *Pattern-oriented* (Sankar & Hejmadi, 2008). Många av dessa ansatser beskrivs bara i enstaka artiklar. Nedan beskrivs några av de mer framträdande ansatserna: GORE, AORE och *pattern-oriented*, samt ansatser kopplade till *Problem frames*, *software product line engineering*, agil systemutveckling, tjänstebaserade system och användarfokusering.

4.1.1 Goal-oriented requirements engineering

För kravhanteringsansatsen GORE utgör de olika mål organisationer och aktörer har grunden till hur krav för ett tänkt system identifieras och hanteras. Utifrån målen fastställs kraven på det system som ska utvecklas (Werneck, Oliveira, & Leite, 2009). De olika aktörernas mål åskådliggörs i målmodeller (eng. goal models) (Alexander, 2011).

Forskning gällande målmodeller handlar bland annat om att transformera intressenters krav till målmodeller (Jureta, Faulkner, & Schobbens, 2008) och att koppla styrande juridiska dokument till mål- och scenariomodeller för att utvärdera om organisationen följer juridiska åtaganden (Ghanavati, Amyot, & Peyton, 2009). Målmodeller bygger på antagandet att alla situationer det tänkta systemet kan hamna i är kända och hanteras under designen av systemet. För självadaptiva system gäller inte detta antagande då alla situationer inte är kända (Whittle, Sawyer, Bencomo, Cheng, & Bruel, 2009). För att hantera detta finns ett komplementärt modelleringsspråk, RELAX, som hanterar problemet med okända situationer (Whittle, Sawyer, Bencomo, Cheng, & Bruel, 2010). Modelleringsramverket *Knowledge Acquisition in autoMated Specification* (KAOS) ses som en möjlighet att hantera adaptiva mål inom målbaserade metoder (Baresi, Pasquale, & Spoletini, 2010).

Inom den målbaserade ansatsen har en rad specifika frågor studerats, som hantering av föränderliga krav (Rohleder, 2008), icke-funktionella krav (Song, Hwong, & Ros, 2009), skillnaden på obligatoriska och valfria mål (Liaskos, McIlraith, Sohrabi, & Mylopoulos, 2010), skalbarhetskrav (Duboc, Letier,

Rosenblum, & Wicks, 2008), liksom kravhantering under upphandlingsprocesser för system baserad på COTS-komponenter (Commercial Off-The-Shelf) (Carvallo & Franch, 2009). Kopplingen mellan verksamheten och kravhantering är något som ett flertal artiklar studerar. Dels relationen mellan målmodeller och verksamhetsprocesser (Snoeck, & Poels, 2009; Rifaut & Dubois, 2008, Grau, Franch, & Maiden, 2008; Windisch, 2009), men även hur verksamhetsmål kan kopplas till kravhantering (Singh & Woo, 2009; Clements & Bass, 2010; Hirzalla, 2010). Då ansatsen baseras på aktörers mål anses den nödvändig att kombinera med metoder som säkerställer användbarhet (Sutcliffe, Thew, & Jarvis, 2011; Patrício, Cunha, & Fisk, 2009).

Metoder inom den målorienterade ansatsen inkluderar olika sätt att upptäcka vilka säkerhetsavvägningar som olika grader av förtroende för aktörer innebär (Elahi & Yu, 2009). Ansatsen inkluderar även verifiering av kravuppfyllnad (Hall, 2009) och bedömning av hur kompletta kraven är (Boness, Finkelstein, & Harrison, 2011).

En rad olika modelleringsramverk stödjer GORE (Werneck, Oliveira, & Leite, 2009). De mer omskrivna är i*, KAOS och TROPOS. Skillnaderna mellan modelleringspråken i* och KAOS diskuteras av Werneck, Oliveira och Leite (2009). Monteiro, Araujo, Amaral och Patrício (2010) diskuterar transformation av modeller mellan dessa två språk med syfte att relatera språkens olika koncept. i* kallas även för ett agentorienterat ramverk (Krishna, Vilkomir, & Ghose, 2009). En agent ses som en entitet med bland annat egna mål varför agent- och målorienterad kravhantering nämns i samband med i* (Moody, Heymans, & Matulevicius, 2010).

Kopplat till i* återfinns studier kring dess visualisering (Moody, Heymans, & Matulevicius, 2010; Horkoff & Yu, 2010; Moody, Heymans, & Matulevicius, 2009), olika former av tillägg och utökningar (Glorio, Pardillo, Mazon, & Trujillo, 2008; Elahi, Yu & Zannone, 2010; Silva, Borba, & Castro, 2011), kvalitetsbedömningar (Cares & Franch, 2009), transformationer mellan i*-varianter (Colomer, López, Cares, & Franch, 2010), transformationer från i*-modeller till andra representationsformer (Decreus, Snoeck, & Poels, 2009; Castro, Lucena, Silva, Alencar, Santos, & Pimentel, 2011) och modellering av kopplingar till andra områden (Lockerbie, Bush, Maiden, Blom, & Everdij, 2010; Nissen, Schmitz, Jarke, Rose, Drews, & Hesseler, 2009; Schmitz, Nissen, Jarke, Rose, Drews, Hesseler, & Reke, 2008). En variant av i* har använts tillsammans med traditionella kravspecifikationer. En styrka hos i* anses vara att det är komplementärt till andra notationer istället för ersättande (Krishna, Vilkomir, & Ghose, 2009). i* har även studerats i relation till modelldriven arkitekturutveckling (Trujillo, Soler, Fernández-Medina, & Piattini, 2009) och modellering av domänmodeller (Nissen, Schmitz, Jarke, Rose, Drews, & Hesseler, 2009). En kritik mot i* är att det fortfarande inte har någon allmänt

accepterad metamodell som styr språket (Colomer, López, Cares, & Franch, 2010).

KAOS har anpassats till adaptiva mål (Baresi, Pasquale, & Spoletini, 2010), samt använts för att representera och dokumentera designbeslut (Gaspar Santos & de Medeiros, 2011). Modelleringsramverket TROPOS är baserat på i* (Werneck, Oliveira, & Leite, 2009) och har utökats med möjligheten att analysera risker under kravanalysfasen (Asnar, Giorgini, & Mylopoulos, 2011). Frankova, Séguran, Gilcher, Trabelsi, Dörflinger och Aiello (2011) utgår från krav modellerade i verktyget Secure TROPOS, baserat på TROPOS metodologi, och transformerar dem till representationer som hanterar verksamhetsprocesser.

4.1.2 Aspect-oriented requirements engineering

AORE erbjuder metoder och tekniker för att hantera krav som är relaterade till varandra men som inte är grupperade i kravspecifikationen, till exempel säkerhetskrav som är utspridda i flera delar av specifikationen (Rashid, 2008). Genom abstrahering och sammansättning kan beroenden mellan krav åskådliggöras i *sammansättningspecifikationer* (eng. composition specification). Detta möjliggör reflektion kring interaktioner och konflikter mellan krav. AORE skiljer sig från GORE genom explicit separation av sammansättnings-specifikationer, vilka tydliggör drivkrafter för arkitektoniska beslut och ger indata till verifikation. För GORE är dessa sammanvävda i målmodeller (Rashid, 2008). För att förbättra AOREs förmåga att identifiera kravkonflikter har ytterligare formalisering av AORE presenterats (Weston, Chitchyan, & Rashid, 2009; Weston, Chitchyan, & Rashid, 2008). För att förenkla analyser finns förslag på att visualisera representationerna grafiskt istället för med text (Oliveira, Araújo, & Amaral, 2010) och en metod för att identifiera konflikter mellan olika aspekter (Mussbacher, Whittle, & Amyot, 2010).

4.1.3 Pattern-oriented baserad kravhantering

Det finns typer av krav som återkommer oavsett vilken sorts system som utvecklas. Genom att definiera kravmönster (eng. requirements patterns) över återkommande krav ges stöd för hur sådana krav ska formuleras och vilken information som ska inkluderas. I litteraturstudien beskrivs hantering av icke-funktionella krav (Supakkul, Hill, Chung, Tun, & Leite, 2010), visualisering av kravmönster (Supakkul & Chung, 2010), hur rätt kravmönster ska väljas (Weiss & Mouratidis, 2008) samt hur flera kravmönster kan aggregeras (Sankar & Hejmadi, 2008).

4.1.4 Problem frames

En viktig egenskap hos ansatsen för problem frames är en tydlig skillnad mellan krav och kravspecifikation (Li, Hall, & Rapanotti, 2011). Krav inom problem frames-ansatsen är något som berör den verkliga världen och förstås av intressenter då det är deras behov som berörs. Kravspecifikationer däremot berör endast systemets funktion. Forskning inom ansatsen problem frames berör transformationen av krav till specifikationer (Dubois & Mouratidis, 2010), identifiering och strukturering av användarinteraktioner (Gervasi, 2010), hur scenarion kan användas inom ansatsen (Chen, Yin, & Jin, 2010) samt att ansatsen kan användas för att hitta fel och dokumentera orsaker till fel i systemet (Tun, Jackson, Laney, Nuseibeh, & Yu, 2009). Lencastre, Moreira, Araújo och Castro (2008) har försökt addera influenser från AORE till problem frames.

4.1.5 Software Product line engineering

När ett antal produkter inom en och samma marknad (domän) har en uppsättning gemensamma funktioner (eng. features), men varje produkt har egna varianter/konfigurationer, benämns det som en produktlinje (eng. product line). Software Product line engineering (SPLE) är uppdelat i två större aktiviteter: *domain engineering* och *application engineering*. *Domain engineering* är en metodik för att dokumentera fakta kring en specifik domän – nyckelkoncept, funktioner, händelser, beteenden, gemensamma och unika funktioner samt deras relationer. I *application engineering* härrörs en specifik produkt utifrån uppsättningen funktioner (Sellier, Mannion, & Mansell, 2008; Mashkoor & Jacquot, 2011). De fördelar med SPLE som lyfts fram mest inom forskningen är ökad *time-to-market* (Kim, Park, & Sugumaran, 2008) och att det är en av de ledande ansatserna inom återanvändning av funktioner (Eriksson, Börstler, & Borg, 2009).

Att arbeta med krav inom SPLE ställer krav på metoder och tekniker, för att kunna hantera en gemensam bas av funktioner och deras relationer, samt de olika produkternas sammansättning av funktioner och säkerställa att de fungerar i sitt sammanhang. Ett krav, eller grupp av krav, kan enligt Eriksson, Börstler och Borg (2009) exempelvis:

- Vara obligatorisk eller valbar, för alla eller endast några produkter i produktlinjen.
- Vara beroende av andra krav för att fungera i systemet.
- Vara uteslutande för andra krav för att fungera i systemet.
- Vara varierande i detaljgraden.

En utmaning med SPLE är att hantera förändringar i både intressenters krav och systemets kontext. Detta område saknar tillfredsställande stöd för att kunna

förtuse och medvetet hantera relaterade förändringar och deras effekter (Peng, Yu, & Zhao, 2011).

Ett antal ansatser för kravhantering inom SPLE har föreslagits, men de anses ännu inte tillräckligt mogna (Alves, Niu, Alves, & Valença, 2010). Det pågår exempelvis forskning för beslutsstöd gällande hur de olika produkterna i produktfamiljen ska sättas samman (Sellier, Mannion, & Mansell, 2008; Classen, Hubaux, & Heymans, 2009), men även forskning kring skapande, verifiering och underhåll av modellerna (Nissen, Schmitz, Jarke, Rose, Drews, & Hesseler, 2009; Mashkooor & Jacquot, 2011; Wang, Zhang, Zhao, Jin, & Mei, 2009; Salinesi, Rolland, Diaz, & Mazo, 2009; Lauenroth & Pohl, 2008; Alturki & Khedri, 2010; Salinesi, Mazo, Diaz, & Djebbi, 2010). Niu och Easterbrook (2008) och Stoiber, Fricker, Jehle och Glinz (2010) menar att vid övergång från en enskild, befintlig produkt till en produktlinje kan det vara resursbesparande att extrahera modeller och dokumentation från det befintliga systemet istället för att genomgå hela processen för domain engineering, i alla fall för att nå en grund för produktlinjen.

4.1.6 Kravhantering inom agil systemutveckling

Kravhantering inom agil systemutveckling skiljer sig åt i jämförelse med mer traditionell kravhantering. I agil systemutveckling föredras ansikte-mot-ansikte kommunikation, och den formella dokumentationen av krav och designbeslut nedprioriteras (Cao & Ramesh, 2008; Rubin & Rubin, 2011). Fördelar med agila ansatser är att beställare får en tydligare kontroll över systemets utveckling, samt att den tid som i traditionell utveckling läggs på dokumenthantering sparas in. En nackdel är att beställare måste leva upp till att hålla en tät kontakt med utvecklaren. Inom beställargruppen måste det också finnas konsensus över vad systemet ska åstadkomma (Cao & Ramesh, 2008).

Iterativ kravhantering i agila miljöer innebär att högnivåkrav för systemet identifieras och antas av utvecklingsteamet i början av ett projekt, men övriga krav utvecklas vartefter systemet skapas. I litteraturstudien inkluderas beskrivningar av gängse praxis inom agil kravhantering (Cao & Ramesh, 2008), vilka gängse praxis kopplat till kravprioritering som har brister (Racheva, Daneva, Sikkell, Wieringa, & Herrmann, 2010), svårigheter med och förslag på sätt att införa agil systemutveckling hos stora organisationer (Savolainen, Kuusela, & Vilavaara, 2010), effekten av den kontinuerliga kontakten med beställare och mindre leveranser (Rubin & Rubin, 2011; Racheva, Daneva, Sikkell, Wieringa, & Herrmann, 2010), samt behovet att studera föränderliga krav (Sahraoui, Buede, & Sage, 2008). Steg för att föra samman automatisk testning och kravhantering i agil systemutveckling har tagits (El-Attar & Miller, 2009; Mugridge, 2008).

Agila systemutvecklingsprocesser ses som *anpassande* efter förutsättningarna snarare än *förutsägande* vilket innebär att systemkod prioriteras framför dokumentering. Men, om dokumenteringen är adaptiv och stödjer samarbete mellan parter snarare än ersätter det, kan specifikationer mycket väl vara en del av den agila processen (Rubin & Rubin, 2011).

4.1.7 Användarfokus

Att skapa värde för intressenterna handlar inte bara om att ta fram nya funktioner, utan främst att förstå deras behov och verksamheter för att kunna hitta medel för att stödja dessa (Kauppinen, Savolainen, Lehtola, Komssi, Töhönen, & Davis, 2009; Gordijn, de Kinderen & Wieringa, 2008). Med en användarfokuserad process kan krav och önskemål hos okända intressenter fångas upp och på så sätt ge utrymme för mer flexibla system som kan vara tilltalande för nya marknader (Kim, In, Baik, Kazman, & Han, 2008) samt att hänsyn kan tas till specifika användares behov (Sun & Mushi, 2010).

Då de flesta som arbetar med kravhantering har en bakgrund inom systemutveckling saknar de ofta kunskap om beteendevetenskap (Fuentes- Fernández, Gómez -Sanz, & Pavón, 2010). Detta anses medföra att de saknar förmågan att korrekt analysera användarnas kontext och identifiera de verkliga behoven. Genom verktygsstöd kan dock samarbetet mellan systemutvecklare, beteendevetare och kunder underlättas.

4.2 Metoder och tekniker

Det finns ett stort antal olika metoder och tekniker att tillgå för att genomföra kravhantering i utvecklingsprojekt. Val av metoder och tekniker beror på projektets ansats, inom vilken kontext systemet ska nyttjas, systemets användare samt kompetensen hos dem som utvecklar systemet.

4.2.1 Identifiera krav

Då identifiering av krav är en hörnsten i kravhantering har det utvecklats en uppsjö av tekniker (Dieste, Juristo, & Shull, 2008) och verktyg vars syfte är att systematisera identifieringen och analysen av intressenters behov (Sim, Alspaugh, & Ban Al-Ani, 2008; Dieste, Lopez, & Ramos, 2008; Carrizo, Dieste, & Juristo, 2008). Emellertid tycks nya ansatser bygga på beprövade tekniker som intervjuer och att skapa prototyper (Sim, Alspaugh, & Ban Al-Ani, 2008; Dieste, Juristo, & Shull, 2008; Dieste, Lopez, & Ramos, 2008). En empirisk lärdom är att effektiviteten i identifieringen av krav ökar då olika varianter av en teknik kombineras, exempelvis två eller flera scenarier som används med samma användare (Seyff, Maiden, Karlsen, Lockerbie, Grünbacher, Graf, & Ncube, 2009), eller kombinationer av olika tekniker, exempelvis intervjuer och att låta

användare tänka högt kring det önskade systemet (Dieste, Juristo, & Shull, 2008). Carrizo, Dieste och Juristo (2008) presenterar ett ramverk för att stödja utvecklare i valet av kravinsamlingstekniker vid utförandet av olika typer av projekt.

Det finns en rad olika metoder och tekniker för att identifiera krav från intressenter (Dieste, Juristo, & Shull, 2008). Några exempel på tekniker är användningsfall (eng. use cases) (Ordinez, Donari, Santos, & Orozco, 2010), scenarier (Seyff, Maiden, Karlsen, Lockerbie, Grünbacher, Graf, & Ncube, 2009; Kaindl, Constantine, Pastor, Sutcliffe, & Zowghi, 2008; Combes, Harel, & Kugler, 2008), *storyboards* (Sutherland & Maiden, 2010; Berenbach & Rayment, 2008) och *card sorting* (Maiden, 2009; Dieste, Juristo, & Shull, 2008). Användningsfall är en av de vanligare teknikerna och i litteraturstudien ges exempel på hur textbaserade användningsfall kombineras med modelldriven systemutveckling (Smith, & Havelund, 2008.) samt hur användningsfall automatiskt kan skapas från textbaserade krav (Seresht & Ormandjieva, 2008). Återanvändning är något som ofta eftersträvas för att underlätta arbetet och befästa den kunskap som finns i organisationen. Kaindl, Smialek och Nowakowski (2010) beskriver en teknik där krav återvinns från tidigare kravspecifikationer. Rosado, Fernández-Medina, López och Piattini (2010) beskriver en teknik där användningsfall kan återvinnas från tidigare projekt.

Krav kan även identifieras från dokument, som organisationens verksamhetsmålsbeskrivning (Herrmann & Paech, 2008), likväl som de automatiskt kan identifieras ur juridiska dokument (Maxwell & Anton, 2009). Det är av vikt att ha metoder som stödjer identifiering från dokument och som möjliggör för intressenter att identifiera och klargöra krav under utvecklingsstegen (Redondo, Ramos-Cabrer, & Gil-Solla, 2009).

Vid projekt för en global marknad finns speciellt behov av metoder som tar hänsyn till intressenter med stor geografisk spridning. Genom att stödja kommunikation med och mellan intressenter (Aranda, Vizcaíno, Cechich, & Piattini, 2008; Fricker, Gorschek, Byman, & Schmidle, 2009), med stödverktyg för samarbete (Lohmann, Heim, & Lauenroth, 2008) och processer för att hantera tusentals intressenter, (Castro-Herrera, Duan, Cleland-Huang, & Mobasher, 2008) kan vanliga problem undvikas.

Genom att framhålla kravidentifiering som kreativ problemlösning kan ett snävt ingenjörsperspektiv undvikas och metoder från andra domäner bli tillgängliga för att stödja skapandet av nya, innovativa lösningar (Maiden, Jones, Karlsen, Neill, Zachos, & Milne, 2010; Nguyen & Shanks, 2009). Att kreativt tänka kring problem anses även kunna leda till mer kompletta krav (Zachos & Maiden, 2008). Lärdomar gällande kreativitet anses kunna nyttjas från områden där det kreativa arbetet varit framträdande, så som filmskapande (Wen, Zhang, Liu & Yang, 2010) och improvisationsteater (Mahaux, Heymans, & Maiden, 2010). Med metoder för gruppdiskussioner som stödjer kreativitet kan problemområdet

utforskas och krav av god kvalitet formuleras (Jones, Lynch, Maiden, & Lindstaedt, 2008; Callele, 2010). Kreativitet kan även användas som grund för att genom spel lära ut nyckelbegrepp inom kravhantering (Smith & Gotel, 2008).

För att sammanfoga intressenters åsikter finns metoder med syfte att hantera utsagor (Agouridas, McKay, Winand, & Pennington, 2008), jämkna ihop flera intressenters perspektiv (Barragáns Martínez, Pazos Arias, Fernández Vilas, García Duque, López Nores, Díaz Redondo, & Blanco Fernández, 2008), representera olika intressenters intressen och inverkan (Ballejos & Montagna, 2011) samt metoder för omhändertagande av mjuka frågor rörande känslor och värderingar (Thew & Sutcliffe, 2008; Callele, Neufeld, & Schneider, 2009b).

4.2.2 Analysera krav

Ett vanligt förekommande problem med kravspecifikationer är att de innehåller en rad felaktigheter (Firesmith (2003). För att komma tillrätta med detta finns både metoder och tekniker som går ut på att bedöma kvalitén på krav (Sustar, Pfeil, & Zaphiris, 2008), identifiera defekter (Vilella, Doerr, & Gross, 2008) eller andra avvikelser i krav. Sådana tekniker kan antingen vara scenariobaserade (Mavin, Novak, Wilkinson, Maiden, & Lynch, 2008; Combes, Harel, & Kugler, 2008; Mussbacher, Whittle, & Amyot, 2009) eller bygga på identifiering av defekter i dokument skrivna i naturligt språk (Tjong & Berry, 2008; Weber-Jahnke & Onabajo, 2009). Det finns även metoder för att utvärdera hela kravhanteringsarbetets resultat, inte utifrån kvalitén på kravspecifikationen, utan utifrån framgången för projektet i helhet (Gorschek & Davis, 2008).

De framtagna kraven behöver prioriteras för att de mest väsentliga funktionerna och egenskaperna ska ingå i systemet. En rad metoder finns beskrivna för hur kravprioritering kan ske utifrån intressenternas åsikter (Barragáns Martínez, Pazos Arias, Fernández Vilas, García Duque, López Nores, Díaz Redondo, & Blanco Fernández, 2008), verksamhetsmål (Beatty, 2010), kostnader (Regnell, Berntsson Svensson, & Olsson, 2008) eller en kombination av intressenter, verksamhetsmål och kvalitetsfaktorer (Duan, Laurent, Cleland-Huang, & Kwiatkowski, 2009).

När kraven identifieras och analyseras måste en reflektion göras om när tillräcklig kravanalys har gjorts för att gå vidare med designen av systemet. Boness, Finkelstein och Harrison (2011) presenterar en metod för att göra en analys för att bedöma förtroendet för ett kravdokument.

Utvecklingen av kravspecifikationer görs genom att ackumulera kunskap om det önskade systemet steg för steg, och redan tidigt i projekt måste avgörande beslut tas. Feather, Cornford, Hicks, Kiper och Menzies (2008) beskriver en beslutsstödmotod som avses stödja beslutsfattande. Detta för att beslut inte endast baseras på tidigare erfarenheter, erfarenheter som kan vara irrelevanta. Al-

Emran, Pfahl och Ruhe (2010) presenterar en beslutsstödsmetod för att bedöma robustheten i krav inför att en ny version av systemet släpps.

Under utvecklingsprocessen förändras ofta krav, och de måste uppdateras. Bittner, Reiser, Glathe och Weber (2010) och Monteiro, Ebert och Recknagel (2009) har tagit fram tekniker för att synkroniserat utbyta krav och kravspecifikationer på ett formaliserat sätt mellan exempelvis kund och leverantör.

4.2.3 Säkerhetskrav

Utvecklingsprojekt tar sällan hänsyn till informationssäkerhet redan från början, vilket de borde (Tøndel, Jaatun, & Meland, 2008). Det finns en rad olika metoder beskrivna som stödjer hanteringen av säkerhetskrav gällande personlig integritet (Kalloniatis, Kavakli, & Gritzalis, 2008; Deng, Wuyts, Scandariato, Preneel, & Joosen, 2011; Vivas, Agudo, & López, 2010), tillträde (He & Antón, 2009), att upptäcka säkerhetsproblem (Houmb, Islam, Knauss, Jürjens, & Schneider, 2010), att ta hänsyn till systemkontexten (Faily & Flechais, 2009) samt identifiera och säkerställa tillförlitligheten hos kärnfunktioner (Kang & Jackson, 2010). Det finns även ramverk som integrerar olika metoder och tekniker för identifiering av säkerhetskrav (Fabian, Gürses, Heisel, Santen, & Schmidt, 2010).

4.2.4 Tjänstebaserade system

System baserade på tjänster levererade av tredje part innebär en rad utmaningar för kravhantering, där en dynamisk och föränderlig driftmiljö utgör ett exempel (Qureshi & Perini, 2010; Elahi, Yu & Zannone, 2010). I litteraturstudien presenteras metoder och verktyg för utformning och utvärdering av kravspecifikationer som berör informationssäkerhet (Morali & Wieringa, 2010; Frankova, Séguran, Gilcher, Trabelsi, Dörflinger, & Aiello, 2011), hur det går att resonera kring kravuppfyllande och att förutspå problem under tiden system är i drift och utvecklas dynamiskt (Elahi, Yu & Zannone, 2010) samt hur kravhantering fortgår under drift genom att kontinuerligt förfina krav i den dynamiska driftmiljön (Qureshi & Perini, 2010).

Det finns tekniker för hur kravhantering kan ske vid kombinerad av e-tjänster (Gordijn, de Kinderen, & Wieringa, 2008), och vid scenario-baserat arbete (Combes, Harel, & Kugler, 2008) för att identifiera oönskade interaktioner mellan befintliga och nya funktioner. Det har även gjorts forskning för att ta fram ett verktyg som, genom att leta fram tjänster på Internet vilka på något sätt kan relateras till det system som ska kravställas, kan ge insikter till nya och innovativa krav och därmed göra kraven mer kompletta (Zachos & Maiden, 2008).

Metoder och tekniker för kravhantering hos tjänstebaserade system har även kombinerats med *service management* (Regev, Hayard, Gause, & Wegmann, 2009), tjänstedesign (Sutherland & Maiden, 2010), interaktionsdesign och marknadsföring (Patrício, Cunha, & Fisk, 2009) samt med verksamhetsprocesser och verksamhetsmål gällande SOA (Liegl, Schuster, Zapletal, Huemer, Werthner, Aigner, Bernauer, Klinger, Mayr, Mizani, & Windisch, 2009; Hirzalla, 2010).

4.2.5 Val av metod och teknik

På grund av den mängd olika metoder och tekniker som finns tillgängliga för olika typer av projekt och kontexter är det problematiskt för kravanalytiker att veta vilka tekniker som bör tillämpas (Jiang, Eberlein, & Far, 2008; Jiang, Eberlein, Far, & Mousavi, 2008; Carrizo, Dieste, & Juristo, 2008). I dagsläget görs valet ofta utifrån personliga erfarenheter vilket inte alltid är optimalt. Det finns därför ett behov av stöd i att välja lämpliga metoder och tekniker (Jiang, Eberlein, & Far, 2008; Jiang, Eberlein, Far, & Mousavi, 2008). Det finns ett antal metoder som stödjer valet av hur arbete ska genomföras (Fabian, Gürses, Heisel, Santen, & Schmidt, 2010) avseende exempelvis kravidentifieringstekniker (Dieste, Lopez, & Ramos, 2008; Carrizo, Dieste, & Juristo, 2008), generella kravhanteringstekniker (Jiang, Eberlein, & Far, 2008; Jiang, Eberlein, Far, & Mousavi, 2008), scenario-baserade tekniker (Hassine, Rilling, & Dssouli, 2010) samt verktyg för att hantera krav (Jiang, Eberlein, Far, & Mousavi, 2008).

Det har även tagits fram metoder som har fokus på en specifik kontext eller förutsättning, så som Bryl, Giorgini och Mylopoulos (2009), som tagit fram en metod för att utforska kravalternativ hos socio-tekniska system. Ytterligare ett exempel är Ameller, Franch, Xavier och Cabot (2010), som inför begreppet icke-funktionella krav i modelldrivet utvecklingsarbete.

4.3 Intressenter och deras krav

Det ingår många olika verksamheter och individer, intressenter (eng. stakeholder), i utvecklingsprojekt. Ofta behöver flera intressenters intressen beaktas vid kravarbetet. Ett sätt att hantera flera olika intressenter är att använda någon form av modellering och intressentprofiler (Castro-Herrera, Cleland-Huang, & Mobasher, 2009; Ballejos & Montagna, 2011; Barragáns Martínez, Pazos Arias, Fernández Vilas, García Duque, López Nores, Díaz Redondo, & Blanco Fernández, 2008; Agouridas, McKay, Winand, & Pennington, 2008; Jureta, Faulkner, & Schobbens, 2008; Finkelstein, Harman, Mansouri, Ren, & Zhang, 2009).

Många utvecklare satsar på att bygga upp mjukvaruplattformar med flera olika applikationer, vilket underlättar återanvändning av både mjukvara och krav, men

skapar andra svårigheter, i synnerhet när det gäller antalet intressenter och olika typer av användare (Song, Hwong, & Ros, 2011). Ett sätt att underlätta kommunikation och samverkan mellan olika intressenter är att använda sig av metoder och verktyg för just detta. Lohmann, Heim och Lauenroth (2008) presenterar ett webbaserat verktyg för att underlätta distribuerad kravidentifiering, Castro-Herrera, Duan, Cleland-Huang och Mobasher (2008) beskriver hur *data mining* metoder kan nyttjas för att hantera processer där väldigt många intressenter är involverade. Ytterligare ansatser för att skapa bättre kommunikation och validering av krav mellan intressenter är att skapa gemensamma språk (Jureta, Mylopoulos, & Faulkner, 2009), använda iterativa och inkrementella processer som tar hänsyn till olika perspektiv (Pires, Delicato, Cóbe, Batista, Davis, & Song, 2011) samt prototyper för validering (Gabrysiak, Giese, & Seibel, 2009).

Ett vanligt problem som uppstår när många intressenter är involverade är att det framkommer krav som är motstridiga (Alexander, 2008; Castro-Herrera, Duan, Cleland-Huang, & Mobasher, 2008). För att hantera detta kan mer eller mindre automatiserade viktningmetoder användas för att ranka och prioritera krav (Zhang, Harman, Finkelstein, Afshin Mansouri, 2011; Alexander, 2008). Alexander (2008) menar att detta sätt saknar matematiskt stöd och föreslår därför istället användandet av mer robusta metoder som lämnar beslutsansvaret hos en människa.

4.3.1 Användare- och verksamhetsbehov

Det är inte bara inom kravhanteringsområdet som krav och behov behandlas. Användarcentrerad design och användbarhetsområdet har stort fokus på att hantera slutanvändarens behov och de krav som det tänkta systemets kontext ställer. Ett flertal artiklar diskuterar möjligheten att kombinera erfarenheter och metoder från dessa områden med kravhanteringsansatser (Sutcliffe, Thew, & Jarvis, 2011; Callele, Neufeld, & Schneider, 2010b; Mukasa & Kaindl, 2008; Kaindl, Constantine, Pastor, Sutcliffe, & Zowghi, 2008; Callele, Neufeld, & Schneider, 2010b; Sustar, Pfeil, & Zaphiris, 2008; Adam, Doerr, Eisenbarth, & Gross, 2009; Heiskari, Kauppinen, Runonen, & Männistö, 2009; Montabert, McCrickard, Winchester, & Pérez-Quñones, 2009; Thew, Sutcliffe, Buijn, McNaught, Procter, Venters, & Buchan, 2008). Ett skäl till detta är att det viktigaste i en utvecklingsprocess bör vara att skapa ett mervärde för den verksamheten som systemet utvecklas för (Kauppinen, Savolainen, Lehtola, Komssi, Töhönen, & Davis, 2009; Gorschek & Davis, 2008; Singh & Woo, 2009). Kravens giltighet bör prövas, inte mot hur väl produkten stämmer överrens med kraven, utan hur väl produkten stämmer överrens med kundens behov i den kontext där den ska användas (Kaindl & Svetinovic, 2010; Kauppinen, Savolainen, Lehtola, Komssi, Töhönen, & Davis, 2009; Faily & Flechais, 2009).

Det är inte säkert att identifierade krav verkligen stämmer överrens med det behov som intressenten har. Krav kan tolkas olika utifrån olika perspektiv, vilket kan skapa problem under kravhanteringsprocessen då olika intressenter har olika tolkningar av samma krav (Kaindl & Svetinovic, 2010). Sätt att hantera detta är modellering och visualisering av krav (Sun & Mushi, 2010; Rohleder, 2008). Detta kan göras för att underlätta utvecklarnas identifiering av krav, men även för att göra det lättare för användarna att förstå olika varianter av icke-funktionella krav (Rohleder, 2008). På det sättet kan användarens mål och förväntningar matchas med produktens krav ur utvecklarnas perspektiv (Rohleder, 2008; Gross & Doerr, 2009). Visualiseringar och modelleringar kan även skapa en större förståelse för kundens behov hos utvecklarna och kravanalytikerna (Dieste, Juristo, & Shull, 2008; Pérez & Valderas, 2009).

Ett nära samarbete mellan kunder och andra intressenter i utvecklingsprocessen ger bättre förutsättningar för en marknadsanpassad produkt (Recio, Salzberg, Palm, & Machuca, 2008). Att kundens och slutanvändarens behov ska finnas med i kravprocessen är viktigt. Ett sätt att åstadkomma detta är att öka medvetenheten om krav i hela processen samt genom förbättrad kommunikation under utvecklingen samt säkerställa att kravdokumenten håller hög kvalitet (Aranda, Vizcaíno, & Piattini, 2010). Ytterligare ett sätt att förbättra samverkan mellan utvecklare och användare är att använda verktyg som låter användarna direkt rapportera in sina behov när de uppstår. Seyff, Graf och Maiden (2010) beskriver ett mobilt verktyg där användare med hjälp av en mobiltelefonapplikation kan föra in behov under sitt dagliga arbete.

4.4 Ontologi

Det råder begreppsförvirring inom kravhanteringsområdet samt bristande interoperabilitet mellan olika modeller och metoder. För att överkomma detta arbetas det fram bland annat ontologier och andra former av begreppsapparater. Inom kravhantering kan ontologier användas för flera olika ändamål (Pires, Delicato, Cóbe, Batista, Davis, & Song, 2011):

- Beskriva en produkt som skapas under kravhantering, exempelvis en beskrivning av kontexten i form av en ontologi.
- Stödja kravhanteringsprocessen, exempelvis integrationen av olika modeller.
- Stödja samverkan inom kravhantering, exempelvis genom att stödja språkbruket under utvecklingen.
- Stödja validering av krav, exempelvis genom att påvisa inkonsistenser.

Eftersom det inom kravhanteringsområdet saknas en konsistent och generellt accepterad begreppsanvändning (Kaindl & Svetinovic, 2010; Fabian, Gürses, Heisel, Santen, & Schmidt, 2010) föreligger risken för missförstånd mellan

utvecklare samt mellan utvecklare och intressenter (Helming, Koegel, Schneider, Haeger, Kaminski, Bruegge, & Berenbach, 2010). Begreppsförvirringen leder till att det är svårt att skapa spårbarhet mellan olika ansatser, metoder och notationer (Helming et al., 2010; Monteiro, Araujo, Amaral, & Patrício, 2010). Forskning inom området berör definition och tydliggörande av begrepps innebörd och betydelse i form av exempelvis ontologier.

Helming et al. (2010) menar att för att möjliggöra modellering av komplexa system krävs en notation som stödjer interdisciplinär samverkan, där begreppen är explicita och tydligt definierade i relation till varandra.

Kaindl och Svetinovic (2010) menar att ett av de problem som ligger till grund för begreppsförvirring inom fältet härrör från att det inte görs någon skillnad mellan verkliga krav och representationer av krav. Dessutom utgör ett alltför stort fokus på representationen av krav att många utvecklare inte upplever sig förstå vad användarna egentligen behöver, vilket leder till illusionen att kraven hela tiden förändras (Kaindl & Svetinovic, 2010).

Helming et al. (2010) beskriver ett ramverk för att modellera komplexa system, i samverkan med domänexperter, vilket innefattar en relativt enkel ontologi som beskriver relationer mellan exempelvis

- *Stakeholder, Goal och Goal Dependency.*
- *Goal, Feature, Functional requirements, Non-functional requirements och Requirements.*

Fabian et al. (2010) beskriver definitioner av begrepp inom kravhantering och deras relationer, med ett tydligt fokus på kravställning av säkerhet. Vidare beskriver och jämför Fabian et al. (2010) flera andra metoder för kravställning av säkerhet, exempelvis Common criteria, SecureTROPOS, SREP, MSRA, samt metoder baserade på UML och Problem frames. I detta arbete beskrivs också hur begrepp inom dessa metoder används. Monteiro et al. (2010) föreslår en ansats för att transformera modeller mellan olika målbaserade metoder. Pires, Delicato, Cóbe, Batista, Davis och Song (2011) beskriver hur ontologier nyttjas för att stödja varje aktivitet i en modellerbaserad ansats.

4.5 Modellering

Inom kravhantering är det viktigt att utvecklare och intressenter har en gemensam uppfattning om det system som ska utveckla. För att öka förståelsen mellan utvecklare och intressenter anses modeller vara ett bra medium. Modeller ger ett visuellt intryck och kan användas för att öka förståelsen mellan parterna (Lubke, Schneider, & Weidlich, 2008; Moody, Heymans, & Matulevicius, 2009). För att underlätta för intressenternas finns även verktyg för att analysera framtagna modeller (Horkoff & Yu, 2010).

Genom att modellera intressenterna kan deras intressen och roller fångas och vara en del av designen av utvecklingsprocessen (Ballejos & Montagna, 2011).

I litteraturen beskrivs ett antal olika modeller. Förutom intressentmodeller är de vanligast förekommande modellerna funktionsmodeller, målmodeller och processmodeller.

Alla typer av modeller lämpar sig inte för alla utvecklingsprojekt och i viss litteratur föreslås kombinationer av olika modeller för att åstadkomma bättre resultat (Laguna & González-Baixauli, 2008; Krishna, Vilkomir, & Ghose, 2009; Fortuna, Werner, & Borges, 2008). Då flertalet olika modeller används är transformeringen mellan de olika modellerna viktig för att underlätta arbetet. Arbeten att försöka skapa mer eller mindre automatiska transformeringar mellan modeller har gjorts (Decreus, Snoeck, & Poels, 2009; Castro, Lucena, Silva, Alencar, Santos, & Pimentel, 2011). Forskning bedrivs inte enbart inom transformering mellan modeller utan även där modeller flyttas mellan olika verktyg (Colomer, López, Cares, & Franch, 2010).

Även om modellbaserade ansatser är aktuellt inom kravhantering är naturligt språk ännu en viktig del. De flesta specifikationer skrivs idag i naturligt språk, men det finns ansatser att förändra detta genom försök att ta fram nya modelleringsspråk för att skapa kravspecifikationer, bland annat har Glinz (2010) tagit fram ett modelleringsspråk för att ersätta naturligt språk i kravspecifikationer. Ansatser att transformera krav skrivna i naturligt språk till modeller finns (Kof, 2010), bland annat för att identifiera felaktigheter i kravspecifikationer. Även transformeringen från scenarion i naturligt språk till modeller görs (Kof, 2008) samt från krav till målmodeller (Jureta, Faulkner, & Schobbens, 2008).

Med den växande komplexiteten i system blir den adaptiva förmågan hos system allt väsentligare. Baresi, Pasquale och Spoletini (2010) beskriver en målmodell som omhändertar just adaptiva egenskaper i system. Att omhänderta krav i realtid är en svårighet, liksom att omhänderta krav som kontinuerligt förändras. Arbeten och modeller har gjorts för att se hur mål som tillkommer och faller bort påverkar befintliga mål (Tanabe, Uno, Akemine, Yoshikawa, Kaiya, & Saeki, 2008). Liknande arbete har även gjorts inom funktionsmodellering (Combes, Harel, & Kugler, 2008). Mindre vanligt beskrivna är de modeller där sekretess och säkerhet omhändertas i olika aspekter (Deng, Wuyts, Scandariato, Preneel, & Joosen, 2011; Elahi & Yu, 2009; Faily & Flechais, 2009). Trots att mängden modeller och ramverk som redan existerar är stor, görs ett flertal försök att anpassa och vidareutveckla dessa ramverk. Detta görs ofta för att brister identifierats, men även för att utöka användningsområdet och få ramverket att fungera för flera kontexter (Wolter & Meinel, 2010; Silva, Borba, & Castro, 2011; Cares & Franch, 2009; Ali, Dalpiaz, & Giorgini, 2010; Helming et al., 2010; Soares, Vrancken, & Verbraeck, 2011).

4.6 Kravdokument och språk

Kravdokument är viktiga för att dokumentera och förmedla krav. Litteraturstudien visar på praktiska erfarenheter av vikten av att etablera en effektiv dokumentstandard, samt att införa gransknings- och revisionsprocesser (Konrad & Gall, 2008; Redondo, Ramos-Cabrera, & Gil-Solla, 2009; Mavin, Wilkinson, Harwood, & Novak, 2009; Mavin & Wilkinson, 2010).

De flesta kravdokument skrivs med naturligt språk. Fördelen med detta är att slippa översätta från det naturliga språket till ett formellt språk, samt att icke-professionella utvecklare har lättare att ta till sig och bedöma krav formulerade på naturligt språk. Eriksson, Börstler och Borg (2009) föreslår en ansats för att skapa kravspecifikationer baserat på naturligt språk. Vid hantering av krav skrivna på naturligt språk är det fördelaktigt om den kravskrivande organisationen och dess individer har ett heuristiskt förhållande till lärande, som då kommer att ackumulera kunskap om tydlig kravskrivning. Detta är ett organisatoriskt förhållningssätt och kan inte automatiseras (Knauss, Schneider, & Stapel, 2009). Arbete med att hantera logisk tvetydighet, det vill säga krav som innehåller tvetydiga ord som ”och” och ”eller”, försiggår (Tjong & Berry, 2008). Tvetydighet som beror på syntaktisk inkonsekvens kan åtgärdas genom att strikt formulera krav enligt fördefinierade syntax-mallar (Mavin, Wilkinson, Harwood, & Novak, 2009; Mavin & Wilkinson, 2010).

Kravspecifikationer innehåller ofta en ordlista för att ensa begrepp och definitioner. För att slippa skapa nya definitioner i nya domäner har Wolter, Smialek, Bildhauer och Kaindl (2008) utvecklat ett nytt språk för kravspecifikationer där nomenklatur återanvänds genom uppslagning av ord från ett lexikon tillgängligt på Internet. För att säkerställa validiteten av artefakter framkomna under kravhanteringen förs en dialog mellan intressenter och utvecklare. Jureta, Mylopoulos och Faulkner (2009) föreslår ett språk för att representera informationsutbytet vid dessa diskussioner mellan intressenter och utvecklare.

Det är ett problem att de flesta kvantitativa krav uttrycks som kvalitativa, till exempel "responstiden måste vara snabb", vilket gör det svårt att tolka och rangordna kraven. Detta brukar åtgärdas genom att kvantifiera dem. Kvantifiering kan i sig orsaka nya problem i form av system som är svagare än vad beställare önskat sig, eller starkare än vad ekonomin tillåter, eller helt enkelt slutar i att utvecklare och beställare inte kan komma överens om en bra kvantifiering av kraven. Glinz (2008) förespråkar en kvantitativ representation genom en risk-baserad strategi.

Formella språk har tagits fram för att lösa problemen med naturligt språk. Siegl, Hielscher och German (2010) förespråkar formella språk, vars fördelar innefattar att kraven blir entydiga och ger möjlighet att skapa relationer som underlättar att utvärdera kraven. Exempelvis Siegl, Hielscher och German (2010) presenterar

ett formellt språk med syfte att undvika tvetydigheter. Vidare så underlättar det formella språket identifiering av saknade eller felaktiga krav samt fel i designen. Navarro, Leveson och Lundqvist (2010) introducerar ett koncept kallat semantisk koppling (eng. semantic coupling). Via spårbarhetsmatriser, vilka representerar kopplingen mellan olika abstraktionsnivåer, kan känsligheten hos en viss design för förändringar i kraven utvärderas.

Stora ansträngningar görs för att utveckla kravmodelleringspråk, för att komma ifrån att majoriteten av kravspecifikationer skrivs baserat på naturligt språk. Svårigheten att få genomslag för dessa anses inte bero på en oförmåga hos kravanalytiker att ta till sig modelleringsmetoder, utan ses snarare som en indikation på att tunga modelleringspråk, såsom UML, inte passar kravanalytikernas behov (Glinz, 2010, Yang, Roeck, Gervasi, Willis, & Nuseibeh, 2011; Kiyavitskaya, Zeni, Mich, & Berry, 2008; Tjong & Berry, 2008; Siegl, Hielscher, & German, 2010, Kamalrudin, Hosking, & Grundy, 2010). Samtidigt menar Sim, Alspaugh och Ban Al-Ani att alltför detaljerade specifikationer inte nödvändigtvis bra, utan att de måste skrivas på en abstraktionsnivå som är anpassad till dess syftet.

Whittle, Sawyer, Bencomo, Cheng och Bruel (2009) har utvecklat ett nytt språk för att hantera kravspecifikationer för adaptiva system. Språket utgörs av olika operatörer som tillåter systemet att omhänderta osäkerheter i systemmiljön. Det medger även att systemet kan välja att vissa icke-kritiska krav kan kopplas från i en nödsituation för att säkerställa att andra kritiska krav istället kan uppfyllas.

4.7 Spårbarhet

Spårbarhet i kravhantering handlar om att relatera krav till sitt ursprung och till de lösningar som medför att kravet uppfylls. Spårbarhet kan finnas mellan krav, modeller, design och färdig kod. Spårbarhet kan innefatta möjlighet att se vilka krav som påverkas vid förändring av ett specifikt krav (Soares, Vrancken, & Verbraeck, 2011), vilken kod som har genererats ifrån vilka krav (Post, Sinz, Merz, Gorges, & Kropf, 2009) och att koppla krav till designelement för att identifiera ouppfyllda krav (Holbrook, Huffman Hayes, & Dekhtyar, 2009). Även olika visuella tekniker används för att påvisa relationer mellan krav (Heim, Lohmann, Lauenroth, & Ziegler, 2008).

Spårbarhet anses vara mest nödvändigt i långlivade mjukvarusystem där utvecklare efterhand förlorar kunskap om källkoden. Spårbarhetslänkar bör lagras som metadata om kravet, exempelvis intressenters namn samt var, när och av vem kravet dokumenterades. Spårbarhet kan bidra med att identifiera förändringar i krav och förutsäga deras påverkan på de senare faserna i utvecklingsprocessen, vilket sänker kostnader kopplade till kravs förändlighet (eng. volatility) (Pires, Delicato, Cobe, Batista, Davis, & Song, 2011; Egyed, Graf, & Grünbacher, 2010).

Trots betydande framsteg inom forskning avseende spårbarhet är utvinnande av länkar, från befintliga krav och implementerad kod, fortfarande en arbetsintensiv manuell aktivitet. Egyed, Graf och Grünbacher (2010) har forskat kring kostnad för att utvinna länkar ur befintligt material, och kan påvisa att manuell länkning går förvånansvärt snabbt även för större kvantiteter av krav. Kvaliteten på länkarna förbättras inte med högre ansträngning, och det är inte svårare att spåra länkar till större och mer komplexa klasser. Befintliga kommersiella verktyg, exempelvis Doors, har stöd för lagring och hantering av länkar, men inte för att utvinna dem (Egyed, Graf, & Grünbacher, 2010). Spårbarhet ingår numera i standarder för utvecklingsmetoder (Egyed, Graf, & Grünbacher, 2010).

Testdriven utveckling är en strategi inom den agila systemutvecklingen där tester tas fram innan koden. Tester används för att härleda kod till krav och design-dokumentation. Krav på spårbarhet i ny kod gör att feedback kan ges snabbt vid förändringar som visar sig gå fel. Testerna talar om vad systemets beteende borde vara. En stor utmaning för ett testdrivet arbetssätt är att utvecklare inte är vana att skriva tester innan de skriver koden. De upplever det som bakvänt, och det kräver disciplin. De flesta organisationer anses vara oförmögna att genomföra denna praxis (Cao & Ramesh, 2008; Rubin & Rubin, 2011).

4.8 Prioritering av krav

Två av de större utmaningarna inom kravhantering är när förutsättningarna för system förändras under utvecklingsarbetet och när okända gränssytor för system måste hanteras. Detta innebär att funktionerna i system under utveckling måste begränsas för att en indirekt kontroll av gränssytor ska erhållas. Att ta avgränsningsbeslut (eng. scoping decisions) samt kontinuerligt bedöma förutsättningarna är därför centralt (Wnuk, Regnell, & Karlsson, 2009; Wnuk, Regnell, & Karlsson, 2008; Wnuk, Callele, & Regnell, 2010).

I projekt med flera intressenter med olika krav används ofta viktning, det vill säga poängsättning av krav efter förutbestämda kriterier, som en metod för att nå beslut om hur viktiga enskilda krav är att tillfredställa. Regnell, Berntsson Svensson och Olsson (2008) presenterar en modell för att hantera prioritering av marknadsdrivna krav efter funktionella aspekter. Barragáns Martínez, Pazos Arias, Fernández Vilas, García Duque, López Nores, Díaz Redondo och Blanco Fernández (2008) presenterar en metod för att prioritera kravmängder från olika intressenter baserat på täckningsgrad och densitet. Ovanstående representerar två av många viktningstrategier som är baserade på matematisk grund (Herrmann & Daneva, 2008; Finkelstein, Harman, Mansouri, Ren, & Zhang, 2009; Zhang, Harman, Finkelstein, Afshin Mansouri, 2011), där kriterier bör vara oberoende dimensioner som kan mätas med olika enheter. Finkelstein, Harman, Mansouri, Ren och Zhang (2009) menar dock att matematiska strategier är otillförlitliga då beslut ofta baseras på en okänd algoritm. Den statistiska tekniken *Principal*

Components Analysis erbjuder riktlinjen: "om du föredrar dessa kriterier bör du gynna dessa kandidater". På så sätt läggs ansvar för viktningen på beslutsfattare och ger en bättre avvägningsprocess för projekt (Alexander, 2008).

I agila projekt är prioritering svårare. Agil prioritering bör göras så att användaren drar nytta av den och leverantören känner att systemet gynnas av den. Problemet ligger i att prioritering utförs i varje utvecklingscykel och baseras på utvecklingscykelns användarrepresentants syn på verksamhetsvärde (Racheva, Daneva, Sikkil, Wieringa, & Herrmann, 2010; Cao & Ramesh, 2008). Fördelen är att verksamhetsvärdet verkligen tas till vara, nackdelen är att systemets förmåga till skalbarhet uteblir (Cao & Ramesh, 2008).

4.9 Utvärdering av krav

Forskning inom området utvärdering av krav omhändertar en mängd olika aspekter. Utvärdering kan göras på modeller, färdiga system, kravspecifikationer eller enskilda krav. Nedan beskrivs några av dessa områden.

Utvärdering av krav kan innebära att hela system verifieras mot den kravspecifikation som upprättats, för att avgöra om systemet uppfyller de uppsatta kraven (Hall, 2009). Utvärdering kan även göras av de kravdokument som är framtagna (Knauss & Boustani, 2008) för att mäta nivån av inkonsekvens i dessa (Barragáns Martínez, Pazos Arias, Fernández Vilas, García Duque, López Nores, Díaz Redondo, & Blanco Fernández, 2008). Ett annat sätt att utvärdera kravdokument är att identifiera vilka krav som ej är uppfyllda (Holbrook, Huffman Hayes, & Dekhtyar, 2009).

Många specifikationer är i dag skrivna på naturligt språk och att identifiera felaktigheter så som motsägelser, beroenden eller tvetydigheter är viktigt (Yang, Roeck, Gervasi, Willis, & Nuseibeh, 2011; Kiyavitskaya, Zeni, Mich, & Berry, 2008). Detta kan göras med stöd av mer eller mindre automatiserade verktyg och genom inspektioner. Inspektioner är ett granskningsförfarande där dokument granskas manuellt mot uppsatta kriterier för att identifiera defekter (Komssi, Kauppinen, Pyhäjärvi, Talvio, & Männistö, 2010). Syftet med att identifiera felaktigheter i dokument är bland annat att skapa modeller med högre kvalitet (Kof, 2008). Metoder för att utvärdera krav kan baseras på ramverk (Cares & Franch, 2009), revisionscykler (Redondo, Ramos-Cabrer, & Gil-Solla, 2009), och mallar för att säkerställa nivån på kravformuleringarna (Mavin & Wilkinson, 2010). Utvärdering kan även inriktas på olika typer av krav, exempelvis säkerhetskrav, sekretesskrav eller icke-funktionella krav (Rohleder, 2008; Ghezzi & Tamburrelli, 2009; Massey, Otto, Hayward, & Antón, 2010).

Även att granska specifikationer omnämns i forskningen. España, Condori-Fernandez, González och Pastor (2009) beskriver utvärderingar av kvaliteten av funktionella kravspecifikationer, med fokus på fullständighet (eng.

completeness) och granularitet (eng. granularity). Menzel, Mueller, Gross och Doerr (2010) beskriver ett protokoll för att granska kravdokument avseende fullständighet. Komssi, Kauppinen, Pyhäjärvi, Talvio och Männistö (2010) menar att inspektion av kravspecifikationer är ett effektivt sätt att öka kvalitén på dessa. Kamalrudin, Hosking och Grundy (2010) beskriver verktyget MaramaAI som används för att hitta inkonsekvenser i krav som baserats på naturligt språk och modeller (en typ av användningsfall).

4.10 Transformation

Transformation inom kravhanteringsområdet handlar i stor utsträckning om att transformera mellan krav skrivna på naturligt respektive formella språk (Meziane, Athanasakis, & Ananiadou, 2008). Det bedrivs även forskning inom transformation mellan i* och andra kravhanteringsramverk eller modelleringsspråk (Castro, Lucena, Silva, Alencar, Santos, & Pimentel, 2011). Litteraturen berör till viss del visualisering av krav utifrån olika intressentperspektiv, exempelvis med stöd av metodiken MultiSpec (Barragáns Martínez, Pazos Arias, Fernández Vilas, García Duque, López Nores, Díaz Redondo, & Blanco Fernández, 2008). Ett exempel inom GORE är där en definition och implementation av målmodellstransformationer mellan kravhanteringsramverken i* och KAOS presenteras (Monteiro, Araujo, Amaral, & Patrício, 2010).

4.11 Releasehantering

Releasehantering innebär att utifrån valda kriterier, exempelvis kostnad, värde för intressenter samt avkastning på investeringar, planera vilka egenskaper/funktioner (eng. features) och krav som ska levereras när. Att välja kriterier och planera funktioner samt leverans benämns även *Roadmapping*. För att vara framgångsrik i releasehantering krävs det att intressenters behov och domäner för produkten är väl kända, och att prioritering mellan egenskaper görs.

Det finns ett stort antal metoder för releaseplanering, där de flesta fokuserar på ett specifikt kravområde (Svahnberg, Gorschek, Feldt, Torkar, Saleem, & Shafique, 2010). Forskning inom releasehantering fokuserar på att ta fram metoder, samt förfina eller komplettera befintliga metoder, bland annat sett till beslutstöd och tidsplanering för att minska projekttiderna (Mc Elroy & Ruhe, 2010; Al-Emran, Pfahl, & Ruhe, 2010; Li, van den Akker, Brinkkemper, & Diepen, 2010; Wnuk, Regnell, & Karlsson, 2008). De metoder som tagits fram bedöms som redo att användas, och majoriteten av dem fungerar även för skräddarsydda produkter (Svahnberg, Gorschek, Feldt, Torkar, Saleem, & Shafique, 2010).

4.12 Verktygsstöd

Under kravhantering hanteras stora mängder information där antalet krav och relationer snabbt växer i större projekt och måste hanteras med stödverktyg. Det finns sedan länge etablerade verktyg, som Doors och IBM Requisite Pro, men mognad av ny teknik, som *data mining* och *machine learning*, möjliggör nya former av verktyg (Castro-Herrera, Cleland-Huang, & Mobasher, 2009; Bittner, Reiser, Glathe, & Weber, 2010; Heim, Lohmann, Lauenroth, & Ziegler, 2008; Castro-Herrera, Duan, Cleland-Huang, & Mobasher, 2008). Forskning inom området verktygsstöd för kravhantering fokuserar främst på automatisering, visualisering och transformation. Verktygsstöd för automatisering och transformation behandlar oftast krav skrivna på naturligt språk (Kiyavitskaya, Zeni, Mich, & Berry, 2008; Breaux, 2009) samt spårbarheten hos krav (Pires, Delicato, Cóbe, Batista, Davis, & Song, 2011; Weber-Jahnke & Onabajo, 2009). Verktygsstöd rörande visualisering berör i huvudsakligen grafbaserade verktyg (Heim, Lohmann, Lauenroth, & Ziegler, 2008) och verktyg för systemavgränsning (eng. scoping) (Wnuk, Regnell, & Karlsson, 2008).

Utöver ovanstående visar litteraturstudien på forskning kring verktygsstöd för *Software Product line engineering* (Sellier, Mannion, & Mansell, 2008), funktionsbaserad kravhantering (Waldmann & Jones, 2009), spårbarhet (Valderas & Pelechano, 2009), scenariobaserad kravhantering (Combes, Harel, & Kugler, 2008), stöd för återanvändning av krav (Monzon, 2008), webbaserade verktyg (Lohmann, Heim, & Lauenroth, 2008), simulatorer (Ferreira, Collofello, Shunk, & Mackulak, 2009) samt verktyg rörande de två specifika kravhanteringsramverken i* (Colomer, López, Cares, & Franch, 2010) och KAOS (Gaspar Santos & de Medeiros, 2011). I mindre utsträckning täcker forskningen även verktygsstöd för riskanalys (Asnar, Giorgini, & Mylopoulos, 2011), global/distribuerad mjukvaruutveckling (Romero, Vizcaíno, & Piattini, 2008), kvalitetssäkring (Liu, 2009), kravevolution (Babar, Cox, Tomic, Bleistein, & Verner, 2008) och utbildning inom kravhantering (Smith & Gotel, 2008).

Omoronyia, Sindre, Roper, Ferguson och Wood (2009) presenterar en ansats för att identifiera spårbarhetslänkar mellan användningsfall, den källkod som implementerar dessa, samt vilken eller vilka utvecklare som är ansvariga för implementationen. Dessa länkar visualiseras sedan i verktygsstödet antingen via ordnade listor eller via spårbarhetsgrafer. Bittner, Reiser, Glathe och Weber (2010) hanterar synkronisering av krav mellan tillverkare och underleverantörer, samt identifierar problem under synkroniseringen. Problemen följs upp med en lösning som presenteras i formatet RIF (Requirements Interchange Format), specialdesignat för att utbyta krav mellan kravdatabaser.

4.13 Automatisering

Forskning inom automatisering berör främst de delar av kravhanteringsprocessen som innefattar naturligt språk (Sinha, Paradkat, Takeuchi, & Nakamura, 2010) och användningsfall (Wang, Zhang, Zhao, Jin, & Mei, 2009). Utöver dessa berörs automatisering av spårbarhet hos krav (Sundaram, Hayes, Dekhtyar, & Holbrook, 2010), automatisering av steg inom scenaribaserad kravhantering samt GORE och modelleringsramverket i* (Alspaugh & Anton, 2008). Exempelvis presenteras ett sätt att automatiskt bibehålla spårbarheten vid uppdatering av användningsfall med hjälp av fördefinierade regler för identifiering av förändringar i relationer mellan användningsfall (Mäder, Gotel, & Philippow, 2008).

4.14 Visualisering

Litteraturstudien visar att forskning rörande visualisering inom kravhanteringsområdet under de senaste åren har fokuserat på beslutsstöd (Al-Emran, Pfahl, & Ruhe, 2010) och systemavgränsningar (Wnuk, Callele, & Regnell, 2010). Visualiseringsstöd för kommunikationsanalys (España, Condori-Fernandez, González, & Pastor, 2009), kravanalys (Boness, Finkelstein, & Harrison, 2011) och spårbarhet (Helming, Koegel, Schneider, Haeger, Kaminski, Bruegge, & Berenbach, 2010) samt effektivare visualisering av användningsfall (Lubke, Sneider, & Weidlich, 2008) och grafbaserade visualiseringsverktyg (Moody, Heymans, & Matulevicius, 2009) berörs också. Winkler (2008) utvärderar existerande visualiseringsstödslösningar för spårbarhet utifrån ett föreslaget ramverk och ger förslag på förbättringar och framtida inriktningar inom området.

4.15 Typer av krav

Den vanligaste kravtypen är funktionella krav (eng. functional requirements). De beskriver vad system ska åstadkomma. Övriga typer av krav brukar med ett samlingsnamn benämnas icke-funktionella krav (eng. non-functional requirements eller quality requirements) (Supakkul & Chung, 2010; Song, Hwong, & Ros, 2011; Herrmann & Paech, 2008). Vissa anser att de ”vanliga” metoderna för kravhantering inte hanterar icke-funktionella krav på ett tillfredställande sätt. Exempelvis Ameller, Franch, Xavier och Cabot (2010) beskriver ett ramverk för att integrera icke-funktionella krav i modellbaserad utveckling. Regnell, Berntsson Svensson och Olsson (2008) presenterar en metod för att bedöma icke-funktionella krav utifrån kostnad och det värde de skapar.

Den typ av icke-funktionella krav som tas upp mest i litteraturstudien är säkerhetskrav (eng. security requirements) (Fabian, Gürses, Heisel, Santen, &

Schmidt, 2010; Vivas, Agudo, & López, 2010). I säkerhetskrav ingår bland annat integritetskrav (eng. privacy requirements) (Kalloniatis, Kavakli, & Gritzalis, 2008; Deng, Wuyts, Scandariato, Preneel, & Joosen, 2011), sekretesskrav (eng. confidentiality requirements) (Morali & Wieringa, 2010) och rättighetskrav (eng. authorisations requirements) (Wolter & Meinel, 2010). Young (2011) beskriver en metod för att extrahera krav avseende integritet från exempelvis styrande dokument. Morali och Wieringa (2010) presenterar en metod för specificering av sekretesskrav. Wolter och Meinel (2010) beskriver en notation för att modellera behörighetskrav i UML baserade på verksamhetsmodeller, för att exempelvis beskriva vem som har rätt att utföra en aktivitet.

Två närbesläktade områden till säkerhetskrav är krav på tillförlitlighet (Liu, Wang, & Zhang, 2010) och trygghet (eng. safety) (Hill & Tilley, 2010). Dessa områden berörs dock inte i någon större omfattning i den kravhanteringslitteratur som ingått i denna litteraturstudie.

I adaptiva system ska krav kunna förändras under drift av systemet, beroende exempelvis förändringar i omgivningen (Whittle, Sawyer, Bencomo, Cheng, & Bruel, 2009; Pimentel, Castro, & Franch, 2011). Till denna typ av system hör de tjänstebaserade system, som ska ha möjlighet att anpassas under drift (Ghezzi & Tamburrelli, 2009). Whittle, Sawyer, Bencomo, Cheng och Bruel (2010) presenterar ett språk utvecklat för att skriva krav för adaptiva system.

En typ av krav som kan bli allt viktigare är emotionella krav (eng. emotional requirements). Dessa berör möjlighet att påverka användaren samt att fånga hur användare känner sig (Callele, Neufeld, & Schneider, 2009b; Callele, Neufeld, & Schneider, 2009a). Närliggande typer av krav är upplevelsekrav (eng. experience requirements) (Callele, Neufeld, & Schneider, 2010b) och kognitiva spelkrav (eng. Cognitive gameplay requirements) (Callele, Neufeld, & Schneider, 2010a).

Vidare berör enstaka artiklar krav så som:

- Krav på immateriella rättigheter (eng. intellectual property rights requirements) (Alspaugh, Asuncion, & Scacchi, 2009)
- Lagkrav (eng. legal requirements) (Breaux, Antón, Boucher, & Dorfman, 2008)
- Krav på skalbarhet (eng. Scalability requirements) (Duboc, Letier, Rosenblum, & Wicks, 2008)
- Arkitekturdrivande krav (eng. Architecturally significant requirements) (de Boer & van Vliet, 2009)

4.16 Tjänster

Tjänster har sedan en tid tillbaka setts som ett sätt att åstadkomma adaptiva applikationer, av löst kopplande system. Detta ställer bland annat krav på att under drift kunna förändra kravbilden för att erhålla förändrade beteenden (Qureshi & Perini, 2010; Hirzalla, 2010). Qureshi och Perini (2010) presenterar en metod för att kravställa tjänstebaserade system, under drift. Liegl, Schuster, Zapletal, Huemer, Werthner, Aigner, Bernauer, Klinger, Mayr, Mizani och Windisch (2009) pekar på att tjänster syftar till att koppla samman verksamhetsprocesser och IT-stöd. Ghezzi och Tamburrelli (2009) presenterar en ansats för att övervaka och bedöma hur väl system baserade på tjänster motsvarar krav, prestanda och tillförlitlighet. Detta under det att systemen anpassar sig till förändringar i omgivningen. Regev, Hayard, Gause och Wegmann (2009) pekar på att det finns en skillnad i språkbruket i det som används inom kravhanteringsområdet och det som används inom *Information Technology Infrastructure Library* (ITIL). Inom ITIL används begreppen *service utilities* och *service warranties*, och för att överkomma detta presenterar Regev, Hayard, Gause och Wegmann (2009) hur metoder inom kravhantering och service management kan överbryggas.

5 Diskussion

Kravhantering är en viktig, om inte den viktigaste, aktiviteten för att åstadkomma system av hög kvalitet. Alla förbättringar av hur kravhantering genomförs får ett direkt genomslag i de system som utvecklas, avseende bland annat hur väl de passar sitt syfte. Emellertid är kravhantering i praktisk systemutveckling ett relativt eftersatt område som kräver mer uppmärksamhet. För att skapa en aktuell översikt över den forskning som sker inom kravhantering syftade arbetet som låg till grund för denna rapport att studera vad som är *state-of-the-art* inom området. För att åstadkomma det genomfördes en litteraturstudie baserat på 375 artiklar och papper skrivna mellan 2008-2011.

Det mest framträdandet i litteraturen som ingår i studien är den uppmärksamhet som ansatsen Goal Oriented Requirements Engineering (GORE) har fått inom kravhanteringsområdet (Van Lamsweerde, 2001). GORE är den ansats som störst andel metoder, tekniker och modelleringsnotationer utvecklats för. Av modelleringsramverken med en grund i GORE är i* det som flest intresserat sig för. Vidareutvecklingen av i* fortgår av flera olika forskargrupper för att täcka olika behov. Ännu saknas i i* en allmänt accepterad metamodel (Franch, Maté, Trujillo, & Cares, 2011). En annan ansats, som inte står i kontrast mot GORE, är Software Product Line Engineering (SPLE) vars syfte är att möjliggöra återanvändning i så hög grad som möjligt, exempelvis avseende krav, design och kod. Inom SPLE byggs mjukvara komponentbaserat där varje unikt system konfigureras för att möta sina specifika krav. För Försvarmakten är möjligheten att återanvända utvecklade funktioner och system av intresse (Hallberg, Pilemalm & Westerdahl, 2008). En positiv effekt av detta skulle bland annat vara kortare utvecklingstider för exempelvis ledningssystem. Många av de studier som rör SPLE bedöms som intressanta för Försvarmakten, även om det kan behövas ytterligare forskning avseende kravhantering inom SPLE innan ansatsen är redo att fullt ut tillämpas praktiskt. Ytterligare nytta bedöms Försvarmakten ha av den forskningen som utförs kring releasehantering, då Hallberg, Pilemalm och Westerdahl (2008) identifierade behov av att förändra leveranstakt och leveransinnehåll på ledningssystem – att bygga systemen mer evolutionärt.

En annan ansats kopplad till återanvändning är *requirements pattern* där syftet är att återanvända krav, eller kunskap därom, mellan utvecklingsprojekt för liknande system (Sankar, & Hejmadi, 2008). Inom Försvarmakten pågår många utvecklingsverksamheter och en önskan finns att skapa samordningsvinster. För uppföljning av hur väl krav på förmåga uppfylls, samt för att kunna motivera införskaffande av olika lösningar är spårbarhet av krav en förutsättning. *Requirements patterns* som även stödjer spårbarhet är således av intresse.

Försvarsmakten har tagit beslut om användandet av MODAF¹ som ramverk. MODAF täcker dock inte alla användningsområden i dagsläget, där det kan nämnas att ramverket saknar metoder för kravhantering. Det skulle därför vara av intresse för Försvarsmakten att integrerar en ansats för kravhantering med modellbaserad förmågeutveckling. En kandidat för detta är GORE då den i likhet med MODAFs metamodelld bland annat hanterar verksamhetsmål. Möjligheterna för en sådan integration kräver dock vidare studier. Ytterligare integrationer med MODAF av intresse är modeller för kommunikation med olika intressenter samt beskrivningar av användningskontext.

Flertalet artiklar beskriver nya metoder och tekniker. Dessa bygger ofta på att integrera två eller flera befintliga ansatser för kravhantering. Dessutom finns det en viss öppenhet för att integrerar befintliga ansatser med metoder och tekniker från andra områden, som exempelvis kreativt tänkande och användarcentrerad systemutveckling. Då antalet metoder och tekniker kan vara svåröverskådligt beskrivs även specifika metoder för hur rätt kravhanteringsmetod kan väljas till specifika projekt. Detta är något Försvarsmakten har ett behov av – att veta vilka metoder och tekniker för kravhantering som är lämpligaste för en given verksamhet.

Agil systemutveckling har länge varit ett område som väckt intresse, framförallt inom mjukvaruutveckling. I sina extremer står agil systemutveckling och kravhantering i rak motsats till varandra. Men så väl inom agil systemutveckling som inom kravhanteringsområdet börjar förespråkare närma sig varandra. Framförallt när det gäller system där den omgivning i vilket system ska användas är oklar och oförutsägbart förändlig. Vid agil utveckling är det viktigt att vid behov kunna påvisa hur systemet ska utformas i form av ”kravspecifikationer” vilket inkluderar utveckling av kritiska system.

Att kravställa system som ska fungera i miljöer där adaptivitet är en viktig förmåga diskuteras inom kravhanteringsområdet. Behovet av en adaptiv förmåga hos system uppstår bland annat hos tjänstebaserade system där driftmiljön är dynamisk. För att hantera dessa system kan sätt att skriva krav så att dessa kan förändras under drift behövas. För detta utvecklas bland annat speciella språk med definierade termer och syntax, så som RELAX (Whittle, Sawyer, Bencomo, Cheng, & Bruel, 2010). Försvarsmakten har under en längre tid önskat åstadkomma mer adaptiva och flexibla system genom att anamma ett tjänstekoncept. Därmed är det av intresse för Försvarsmakten att hitta metoder och verktyg som stödjer kravställande av tjänstebaserade system med förmåga att anpassa sig till förändrade förutsättningar.

Ett område som är väl omskrivet berör hantering av säkerhetskrav, avseende exempelvis integritetskrav, sekretesskrav och rättighetskrav (Deng, Wuyts,

¹ FM HKV Beteckning: 09 100:58386 Underbilaga 1.4

Scandariato, Preneel, & Joosen, 2011). I många fall handlar det om metoder och tekniker för hur dessa typer av krav ska identifieras och hur det säkerställs att dessa tillgodoses. Detta innefattar även hur säkerhetskrav ska utvinnas ur styrande dokument. Att hantera säkerhetskrav är högtintressant för Försvarsmakten. Dels avseende hur säkerhetskrav ska identifieras men också för att påvisa att innehållet i styrande dokument får genomslag i systemsäkerheten.

När det gäller verktygsstöd så verkar det saknas verktyg som stödjer samtliga kravhanteringsmoment. Den forskning som beskrivs påvisar möjlighet för att nyttja verktygsstöd för spårbarhet, kravevolution, utbildning inom kravhantering samt synkronisering av krav mellan olika kravdatabaser. Detta finns det behov av inom Försvarsmakten (Hallberg, Pilemalm & Westerdahl, 2008). Det kanske mest intressanta för Försvarsmakten är att nyttja verktyg för utbildning inom kravhantering i form av spel (Smith & Gotel, 2008).

För kravspecifikationer är naturligt språk det vanligaste representationssättet, trots de svårigheter det medför i form av tvetydigheter och motsägelser. Alternativ till naturligt språk finns i form av formella språk, men även dessa medför problem. Framförallt begränsas möjligheten att involvera personer som saknar expertis i det valda formella språket. Därtill finns alternativet att beskriva krav med hjälp av ett flertal olika ramverk med modelleringsnotationer. Genom olika former av visualisering av krav som ett beslutsstöd kan personer utan kravhanteringskompetens göras delaktiga i systemutveckling. Detta innefattar även visualisering av spårbarhet och är av intresse för Försvarsmakten.

Slutsatsen av studien är att det finns mycket för Försvarsmakten att inhämta från kravhanteringsområdets överflöd av metoder och tekniker. Den stora utmaningen är att hitta och anpassa det som motsvarar Försvarsmaktens förutsättningar och vilja att utveckla sina system och förmågor. Vilket i sin tur kräver en vidare forskning.

6 Referenser

- Adam, S., Doerr, J., Eisenbarth, M., & Gross, A. (2009). Using Task-oriented Requirements Engineering in Different Domains – Experiences with Application in Research and Industry. *17th IEEE International Requirements Engineering Conference*, pp. 267-272.
- Agouridas, V., McKay, A., Winand, H., & Pennington, A. (2007). Advanced product planning: a comprehensive process for systemic definition of new product requirements. *Requirements Engineering*, 13(1), 19-48.
- Al-Emran, A., Pfahl, D., & Ruhe, G. (2010). Decision Support for Product Release Planning Based on Robustness Analysis. *18th IEEE International Requirements Engineering Conference*, pp. 157-166.
- Alexander, I. (2008). Evaluating Design Options against Requirements: How Far Can Statistics Help? *16th IEEE International Requirements Engineering Conference*, pp. 259-264.
- Alexander, I. (2011). GORE, SORE, or What? *IEEE Software*, 28(1), 8–10.
- Ali, R., Dalpiaz, F., & Giorgini, P. (2010). A goal-based framework for contextual requirements modeling and analysis. *Requirements Engineering*, 15(4), 439-458.
- Alspaugh, T.A. & Anton, A.I. (2008). Scenario support for effective requirements. *Information and Software Technology*, 50(3), 198-220.
- Alspaugh, T.A. Asuncion, H.U., & Scacchi, W. (2009). Intellectual Property Rights Requirements for Heterogeneously-Licensed Systems. *17th IEEE International Requirements Engineering Conference*, pp. 24-33.
- Alturki, F. & Khedri, R. (2010). A Tool for Formal Feature Modeling Based on BDDs and Product Families Algebra. *13th Workshop on Requirements Engineering*, pp. 109-120.
- Alves, V., Niu, N., Alves, C., & Valença, G. (2010). Requirements engineering for software product lines: A systematic literature review. *Information and Software Technology*, 52(8), 806-820.
- Ameller, D., Franch, Xavier, & Cabot, J. (2010). Dealing with Non-Functional Requirements in Model-Driven Development. *18th IEEE International Requirements Engineering Conference*, pp. 189-198.
- Aranda, G.N., Vizcaíno, A., Cechich, A., & Piattini, M. (2008). A Methodology for Reducing Geographical Dispersion Problems during Global Requirements Elicitation. *11th Workshop on Requirements Engineering*, pp. 117-127.

- Aranda, Gabriela N., Vizcaíno, A., & Piattini, M. (2010). A framework to improve communication during the requirements elicitation process in GSD projects. *Requirements Engineering*, 15(4), 397-417.
- Asnar, Y., Giorgini, P., & Mylopoulos, J. (2010). Goal-driven risk assessment in requirements engineering. *Requirements Engineering*, 16(2), 101-116.
- Babar, A., Cox, K., Tosic, V., Bleistein, S., & Verner, J. (2008). Integrating B-SCP and MAP to manage the evolution of strategic IT requirements. *Information and Software Technology*, 50(7-8), 815-831.
- Ballejos, L.C. & Montagna, J.M. (2011). Modeling stakeholders for information systems design processes. *Requirements Engineering*.
- Baresi, L., Pasquale, L., & Spoletini, P. (2010). Fuzzy Goals for Requirements-Driven Adaptation. *18th IEEE International Requirements Engineering Conference*, pp. 125-134.
- Barragáns Martínez, A.B., Pazos Arias, J.J., Fernández Vilas, A., García Duque, G., López Nores, M., Díaz Redondo, R.P., & Blanco Fernández, Y. (2008). On the interplay between inconsistency and incompleteness in multi-perspective requirements specifications. *Information and Software Technology*, 50(4), 296-321.
- Barragáns Martínez, A.B., Pazos Arias, J.J., Fernández Vilas, A., García Duque, J., López Nores, M., Díaz Redondo, R.P., & Blanco Fernández, Y. (2008). Composing requirements specifications from multiple prioritized sources. *Requirements Engineering*, 13(3), 187-206.
- Beatty, J. (2010). If You Build It, Will They Use It? Leveraging Business Objectives to Deliver Successful Projects. *18th IEEE International Requirements Engineering Conference*, pp. 413-414.
- Berenbach, B. & Rayment, T. (2008). The Evaluation of a Requirements Engineering Training Program at Siemens. *16th IEEE International Requirements Engineering Conference*, pp. 205-210.
- Bittner, M., Reiser, M.-O., Glathe, H., & Weber, M. (2010). Manufacturer-Supplier Requirements Synchronization Using Exchange Containers and Multi-Level Systems. *18th IEEE International Requirements Engineering Conference*, pp. 325-330.
- Boehm, B.W. & Papaccio, P.N. (1998). Understanding and controlling software costs. *IEEE Transaction on Software Engineering*, 14, pp. 1462 – 1477.
- Boness, K., Finkelstein, A., & Harrison, R. (2011). A method for assessing confidence in requirements analysis. *Information and Software Technology*, 53(10), 1084-1096.

- Breaux, T. (2009). Exercising Due Diligence in Legal Requirements Acquisition: A Tool-supported, Frame-Based Approach. *17th IEEE International Requirements Engineering Conference*, pp. 225-230.
- Breaux, T.D., Antón, A.I., Boucher, K., & Dorfman, M. (2008). Legal Requirements, Compliance and Practice: An Industry Case Study in Accessibility. *16th IEEE International Requirements Engineering Conference*, pp. 43-52.
- Bryl, V., Giorgini, P., & Mylopoulos, J. (2009). Designing socio-technical systems: from stakeholder goals to social networks. *Requirements Engineering*, 14(1), 47-70.
- Callele, D. (2010). Physualization: Going beyond paper prototyping. *Fifth International Workshop on Requirements Engineering Visualization*, pp. 35-39.
- Callele, D., Neufeld, E., & Schneider, K. (2009a). Augmenting Emotional Requirements with Emotion Markers and Emotion Prototypes. *17th IEEE International Requirements Engineering Conference*, pp. 373-374.
- Callele, D., Neufeld, E., & Schneider, K. (2009b). Visualizing Emotional Requirements. *Fourth International Workshop on Requirements Engineering Visualization*, pp. 1-10.
- Callele, D., Neufeld, E., & Schneider, K. (2010a). A proposal for cognitive gameplay requirements., *Fifth International Workshop on Requirements Engineering Visualization*, pp. 43-52.
- Callele, D., Neufeld, E., & Schneider, K. (2010b). An Introduction to Experience Requirements. *18th IEEE International Requirements Engineering Conference*, pp. 395-396.
- Cao, L. & Ramesh, B. (2008). Agile requirements engineering practices: An empirical study. *IEEE Software*, 25(1), 60-67.
- Cares, C. & Franch, X. (2009). Towards a Framework for Improving Goal-Oriented Requirement Models Quality. *12th Workshop on Requirements Engineering, WER2009*.
- Carrizo, D., Dieste, O., & Juristo, N. (2008). Study of elicitation techniques adequacy. *11th Workshop on Requirements Engineering*, pp. 104-114.
- Carvalho, J.P., & Franch, X. (2009). On the Use of Requirements for Driving Call-for-Tender Processes for Procuring Coarse-grained OTS Components. *17th IEEE International Requirements Engineering Conference*, pp. 287-292.
- Castro, J., Lucena, M., Silva, C., Alencar, F., Santos, E., & Pimentel, J. (2011). Changing Attitudes Towards the Generation of Architectural Models. *Journal of Systems and Software*. doi:10.1016/j.jss.2011.05.047.

- Castro-Herrera, C., Cleland-Huang, J., & Mobasher, B. (2009). Enhancing Stakeholder Profiles to Improve Recommendations in Online Requirements Elicitation. *17th IEEE International Requirements Engineering Conference*, pp. 37-46.
- Castro-Herrera, C., Duan, C., Cleland-Huang, J., & Mobasher, B. (2008). Using Data Mining and Recommender Systems to Facilitate Large-Scale, Open, and Inclusive Requirements Elicitation Processes. *16th IEEE International Requirements Engineering Conference*, pp. 165-168.
- Chen, X., Yin, B., & Jin, Z. (2010). Dptool: A Tool for Supporting the Problem Description and Projection. *18th IEEE International Requirements Engineering Conference*, pp. 401-402.
- Classen, A., Hubaux, A., & Heymans, P. (2009). Analysis of Feature Configuration Workflows. *17th IEEE International Requirements Engineering Conference*, pp. 381-382.
- Clements, P. & Bass, L. (2010). Using Business Goals to Inform a Software Architecture. *18th IEEE International Requirements Engineering Conference*, pp. 69-78.
- Colomer, D., López, L., Cares, C., & Franch, X. (2011). Model Interchange and Tool Interoperability in the i* Framework: A Proof of Concept. *14th Workshop on Requirements Engineering*, pp. 1-3.
- Combes, P., Harel, D., & Kugler, H. (2008). Modeling and verification of a telecommunication application using live sequence charts and the Play-Engine tool. *Software & Systems Modeling*, 7(2), 157-175.
- de Boer, R.C. & van Vliet, H. (2009). On the similarity between requirements and architecture. *Journal of Systems and Software*, 82(3), 544-550.
- Decreus, K., Snoeck, M., & Poels, G. (2009). Practical Challenges for Methods Transforming i* Goal Models into Business Process Models. *17th IEEE International Requirements Engineering Conference*, pp. 15-23.
- Deng, M., Wuyts, K., Scandariato, R., Preneel, B., & Joosen, W. (2011). A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, 16(1), 3-32.
- Dieste, O., Juristo, N., & Shull, F. (2008). Understanding the customer: what do we know about requirements elicitation? *IEEE Software*, 25(2), 11-13.
- Dieste, O., Lopez, M., & Ramos, F. (2008). Updating a Systematic Review about Selection of Software Requirements Elicitation Techniques. *11th Workshop on Requirements Engineering*, pp. 96-103.

- Duan, C., Laurent, P., Cleland-Huang, J., & Kwiatkowski, C. (2009). Towards automated requirements prioritization and triage. *Requirements Engineering*, 14(2), 73-89.
- Duboc, L., Letier, E., Rosenblum, D. S., & Wicks, T. (2008). A Case Study in Eliciting Scalability Requirements. *16th IEEE International Requirements Engineering Conference*, pp. 247-252.
- Dubois, E. & Mouratidis, H. (2010). Guest editorial: security requirements engineering: past, present and future. *Requirements Engineering*, 15(1), 1-5.
- Egyed, A., Graf, F., & Grünbacher, P. (2010). Effort and Quality of Recovering Requirements-to-Code Traces: Two Exploratory Experiments. *18th IEEE International Requirements Engineering Conference*, pp. 221-230.
- Elahi, G. & Yu, E. (2009). Trust Trade-off Analysis for Security Requirements Engineering. *17th IEEE International Requirements Engineering Conference*, pp. 243-248.
- Elahi, G., Yu, E., & Zannone, N. (2010). A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities. *Requirements Engineering*, 15(1), 41-62.
- El-Attar, M. & Miller, J. (2009). Developing comprehensive acceptance tests from use cases and robustness diagrams. *Requirements Engineering*, 15(3), 285-306.
- Eriksson, M., Börstler, J., & Borg, K. (2009). Managing requirements specifications for product lines – An approach and industry case study. *Journal of Systems and Software*, 82(3), 435-447. Elsevier Inc.
- España, S., Condori-Fernandez, N., Gonzalez, A., & Pastor, Ó. (2009). Evaluating the Completeness and Granularity of Functional Requirements Specifications: A Controlled Experiment. *17th IEEE International Requirements Engineering Conference*, pp. 161-170.
- Fabian, B., Gürses, S., Heisel, M., Santen, T., & Schmidt, H. (2010). A comparison of security requirements engineering methods. *Requirements Engineering*, 15(1), 7-40.
- Faily, S. & Flechais, I. (2009). Context-Sensitive Requirements and Risk Management with IRIS. *17th IEEE International Requirements Engineering Conference*, pp. 379-380.
- Feather, M. S., Cornford, S. L., Hicks, K. A., Kiper, J. D., & Menzies, T. (2008). A broad, quantitative model for making early requirements decisions. *IEEE Software*, 25(April), 49–56. IEEE Computer Society.
- Ferreira, S., Collofello, J., Shunk, D., & Mackulak, G. (2010). Understanding the effects of requirements volatility in software engineering by using analytical

- modeling and software process simulation. *Journal of Systems and Software*, 82(10), 1568-1577. Elsevier Inc.
- Finkelstein, A., Harman, M., Mansouri, S. A., Ren, J., & Zhang, Y. (2009). A search based approach to fairness analysis in requirement assignments to aid negotiation, mediation and decision making. *Requirements Engineering*, 14(4), 231-245.
- Firesmith, D. (2003) Specifying Good Requirements. *Journal of Object Technology*, 2(4), 177-188.
- Fortuna, M. H., Werner, C. M. L., & Borges, M. R. S. (2008). Info Cases: Integrating Use Cases and Domain Models. *16th IEEE International Requirements Engineering Conference*, pp. 81-84.
- Franch, X., Maté, A., Trujillo, J.C., & Cares, C. (2011). On the Joint Use of i* with Other Modelling Frameworks: A Vision Paper. *19th IEEE International Requirements Engineering Conference*, pp. 133-142.
- Frankova, G., Séguran, M., Gilcher, F., Trabelsi, S., Dörflinger, J., & Aiello, M. (2011). Deriving business processes with service level agreements from early requirements. *Journal of Systems and Software*, 84(8), 1351-1363. Elsevier Inc.
- Fricker, S., & Glinz, Martin. (2010). Comparison of Requirements Hand-off, Analysis, and Negotiation: Case Study. *18th IEEE International Requirements Engineering Conference*, pp. 167-176.
- Fricker, S., Gorschek, Tony, Byman, C., & Schmidle, A. (2009). Handshaking: Negotiate to Provoke the Right Understanding of Requirements. *IEEE Software*, 26(99), 1-1.
- Fuentes-Fernández, R., Gómez-Sanz, J. J., & Pavón, J. (2010). Understanding the human context in requirements elicitation. *Requirements Engineering*, 15(3), 267-283.
- Gabrysiak, G., Giese, H., & Seibel, A. (2009). Interactive Visualization for Elicitation and Validation of Requirements with Scenario-Based Prototyping. *2009 Fourth International Workshop on Requirements Engineering Visualization*, 41-45.
- García-Duque, J., Pazos-Arias, J. J., López-Nores, M., Blanco-Fernández, Y., Fernández-Vilas, A., Díaz-Redondo, R. P., Ramos-Cabrer, M., & Gil-Solla, A. (2009). Methodologies to evolve formal specifications through refinement and retrenchment in an analysis–revision cycle. *Requirements Engineering*, 14(3), 129-153.
- Gaspar Santos & Pereira de Medeiros (2011). Design Rationale Representation in Requirements Engineering using the KAOS meta-model. *14th Workshop on Requirements Engineering*, pp 383-394

- Gervasi, V. (2010). Who framed Roger User? Problem frames as a User Interaction design tool. *18th IEEE International Requirements Engineering Conference*, pp. 377-378.
- Ghanavati, S., Amyot, D., & Peyton, L. (2009). Compliance Analysis Based on a Goal-oriented Requirement Language Evaluation Methodology. *17th IEEE International Requirements Engineering Conference*, pp. 133-142.
- Ghezzi, C., & Tamburrelli, G. (2009). Reasoning on Non-Functional Requirements for Integrated Services. *17th IEEE International Requirements Engineering Conference*, pp. 69-78.
- Glinz, M. (2008). A risk-based, value-oriented approach to quality requirements. *IEEE Software*, 25(2), 34-41.
- Glinz, M. (2010). Very Lightweight Requirements Modeling. *18th IEEE International Requirements Engineering Conference*, pp. 385-386.
- Glorio, O., Pardillo, J., Mazon, J.-N., & Trujillo, J. (2008). DaWaRA: An Eclipse Plugin for Using i* on Data Warehouse Requirement Analysis. *16th IEEE International Requirements Engineering Conference*, pp. 317-318.
- González, A., Ruiz, M., España, S., & Pastor, Ó. (2008) Message Structures - a modelling technique for information systems analysis and design. *11th Workshop on Requirements Engineering*, pp. 407-418.
- Gordijn, J., Kinderen, S.D., & Wieringa, R. (2008). Value-driven Service Matching. *16th IEEE International Requirements Engineering Conference*, pp. 67-70.
- Gorschek, T, & Davis, A. (2008). Requirements engineering: In search of the dependent variables. *Information and Software Technology*, 50(1-2), 67-75.
- Grau, G., Franch, X., & Maiden, N. (2008). PRiM: An i*-based process reengineering method for information systems specification. *Information and Software Technology*, 50(1-2), 76-100.
- Gross, A. & Doerr, J. (2009). EPC vs. UML Activity Diagram - Two Experiments Examining their Usefulness for Requirements Engineering. *17th IEEE International Requirements Engineering Conference*, pp. 47-56.
- Hall, R.J. (2009). Forensic System Verification. *17th IEEE International Requirements Engineering Conference*, pp. 111-120.
- Hallberg, N., Pilemalm, S., & Westerdahl, L. (2008) *Behovsanalys avseende Försvarsmaktens utveckling av ledningssystem*. FOI Memo 2443.
- Hallberg, N., Timpka, T., and Eriksson, H. (1999) The Medical Software Quality Deployment Method. *Methods of Information in Medicine*. 38 (1), pp. 66-73.

- Hassine, J., Rilling, J., & Dssouli, R. (2010). An evaluation of timed scenario notations. *Journal of Systems and Software*, 83(2), 326-350. Elsevier Inc.
- He, Q., & Antón, A.I. (2009). Requirements-based Access Control Analysis and Policy Specification (ReCAPS). *Information and Software Technology*, 51(6), 993-1009.
- Heim, P., Lohmann, S., Lauenroth, K., & Ziegler, J. (2008). Graph-based Visualization of Requirements Relationships. *Requirements Engineering Visualization*, pp. 51-55.
- Heiskari, J., Kauppinen, M., Runonen, M., & Männistö, T. (2009). Bridging the Gap Between Usability and Requirements Engineering. *17th IEEE International Requirements Engineering Conference*, pp. 303-308.
- Helming, J., Koegel, M., Schneider, F., Haeger, M., Kaminski, C., Bruegge, B., & Berenbach, B. (2010). Towards a unified requirements modeling language. *Fifth International Workshop on Requirements Engineering Visualization*, pp. 53-57.
- Herrmann, A. & Paech, B. (2008). MOQARE: misuse-oriented quality requirements engineering. *Requirements Engineering*, 13(1), 73-86.
- Herrmann, A., & Daneva, M. (2008). Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research. *16th IEEE International Requirements Engineering Conference*, pp. 125-134.
- Hill, J. & Tilley, S. (2010). Creating Safety Requirements Traceability for Assuring and Recertifying Legacy Safety-Critical Systems. *18th IEEE International Requirements Engineering Conference*, pp. 297-302.
- Hirzalla, M. (2010). Realizing Business Agility Requirements through SOA and Cloud Computing. *18th IEEE International Requirements Engineering Conference*, pp. 379-380.
- Holbrook, E.A., Huffman Hayes, J., & Dekhtyar, A. (2009). Toward Automating Requirements Satisfaction Assessment. *17th IEEE International Requirements Engineering Conference*, pp. 149-158.
- Horkoff, J. & Yu, E. (2010). Visualizations to Support Interactive Goal Model Analysis. *Fifth International Workshop on Requirements Engineering Visualization*, pp. 1-10.
- Houmb, S. H., Islam, S., Knauss, E., Jürjens, J., & Schneider, K. (2010). Eliciting security requirements and tracing them to design: an integration of Common Criteria, heuristics, and UMLsec. *Requirements Engineering*, 15(1), 63-93.
- Jackson, M. (2001). *Problem Frames: Analysing and Structuring Software Development Problems*. New York, Addison-Wesley.

- Jantunen, S., Gause, D. C., & Wessman, R. (2010). Making Sense of Product Requirements. *18th IEEE International Requirements Engineering Conference*, pp. 89-92.
- Jiang, L., Eberlein, A., & Far, B. H. (2008). A case study validation of a knowledge-based approach for the selection of requirements engineering techniques. *Requirements Engineering*, 13(2), 117-146.
- Jiang, L., Eberlein, A., Far, B. H., & Mousavi, M. (2008). A methodology for the selection of requirements engineering techniques. *Software & Systems Modeling*, 7(3), 303-328.
- Jones, S., Lynch, P., Maiden, N., & Lindstaedt, S. (2008). Use and Influence of Creative Ideas and Requirements for a Work-Integrated Learning System. *16th IEEE International Requirements Engineering Conference*, pp. 289-294.
- Jureta, I., Mylopoulos, J., & Faulkner, S. (2009). Analysis of Multi-Party Agreement in Requirements Validation. *17th IEEE International Requirements Engineering Conference*, pp. 57-66.
- Jureta, I.J., Faulkner, S., & Schobbens, P.-Y. (2008). Clear justification of modeling decisions for goal-oriented requirements engineering. *Requirements Engineering*, 13(2), 87-115.
- Kaindl, H. & Svetinovic, D. (2010). On confusion between requirements and their representations. *Requirements Engineering*, 15(3), 307-311.
- Kaindl, H., Constantine, L., Pastor, O., Sutcliffe, A., & Zowghi, D. (2008). How to Combine Requirements Engineering and Interaction Design? *16th IEEE International Requirements Engineering Conference*, pp. 299-301.
- Kaindl, H., Smialek, M., & Nowakowski, W. (2010). Case-based Reuse with Partial Requirements Specifications. *18th IEEE International Requirements Engineering Conference*, pp. 399-400.
- Kalloniatis, C., Kavakli, E., & Gritzalis, S. (2008). Addressing privacy requirements in system design: the PriS method. *Requirements Engineering*, 13(3), 241-255.
- Kamalrudin, M., Hosking, J., & Grundy, J. (2010). MaramaAI: Automated and Visual Approach for Inconsistency Checking of Requirements. *18th IEEE International Requirements Engineering Conference*, pp. 393-394.
- Kang, E. & Jackson, D. (2010). Dependability Arguments with Trusted Bases. *18th IEEE International Requirements Engineering Conference*, pp. 262-271.
- Kasser, J.E. (2007). *A Framework for Understanding of Systems Engineering*. Cranfield, UK: BookSurge Publishing.

- Kauppinen, M., Savolainen, J., Lehtola, L., Komssi, M., Töhönen, H., & Davis, A. (2009). From Feature Development to Customer Value Creation. *17th IEEE International Requirements Engineering Conference*, pp. 275-280.
- Kim, J., Park, S., & Sugumaran, V. (2008). DRAMA: A framework for domain requirements analysis and modeling architectures in software product lines. *Journal of Systems and Software*, 81(1), 37-55.
- Kim, S., In, H.P., Baik, J., Kazman, R., & Han, K. (2008). VIRE: Sailing a Blue Ocean with Value- Innovative Requirements. *IEEE Software*, 25(1), 80–87.
- Kitchenham, B. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. EBSE Technical Report, Keele University, UK.
- Kiyavitskaya, N., Zeni, N., Mich, L., & Berry, D.M. (2008). Requirements for tools for ambiguity identification and measurement in natural language requirements specifications. *Requirements Engineering*, 13(3), 207-239.
- Knauss, E. & Boustani, C. E. (2008). Assessing the Quality of Software Requirements Specifications. *16th IEEE International Requirements Engineering Conference*, pp. 341-342.
- Knauss, E., Schneider, K., & Stapel, K. (2009). Learning to Write Better Requirements through Heuristic Critiques. *17th IEEE International Requirements Engineering Conference*, pp. 387-388.
- Kof, L. (2008). From Textual Scenarios to Message Sequence Charts: Inclusion of Condition Generation and Actor Extraction. *16th IEEE International Requirements Engineering Conference*, pp. 331-332.
- Kof, L. (2010). From Requirements Documents to System Models: A Tool for Interactive Semi-Automatic Translation. *18th IEEE International Requirements Engineering Conference*, pp. 391-392.
- Komssi, M., Kauppinen, M., Pyhäjärvi, M., Talvio, J., & Männistö, T. (2010). Persuading Software Development Teams to Document Inspections: Success Factors and Challenges in Practice. *18th IEEE International Requirements Engineering Conference*, pp. 283-288.
- Konrad, S. & Gall, M. (2008). Requirements Engineering in the Development of Large-Scale Systems. *16th IEEE International Requirements Engineering Conference*, pp. 217-222.
- Kotonya, G. & Sommerville, I. (1998). *Requirements Engineering. Processes and Techniques*, John Wiley & Sons, Chichester.
- Krishna, A., Vilkomir, S., & Ghose, A. (2009). Consistency preserving co-evolution of formal specifications and agent-oriented conceptual models. *Information and Software Technology*, 51(2), 478-496. Elsevier B.V.

- Laguna, M.A. & González-Baixauli, B. (2008). Product Line Requirements: Multi-Paradigm Variability Models. *11th Workshop on Requirements Engineering*, pp. 211-216.
- Lauenroth, K. & Pohl, K. (2008). Dynamic Consistency Checking of Domain Requirements in Product Line Engineering. *16th IEEE International Requirements Engineering Conference*, pp. 193-202.
- Lauesen, S. (2002). *Software requirements: styles and techniques*, Addison-Wesley.
- Lencastre, M., Moreira, A., Araújo, J., & Castro, J. (2008). Aspects Composition in Problem Frames. *16th IEEE International Requirements Engineering Conference*, pp. 343-344.
- Levy Siqueira, F. & Muniz Silva, P.S. (2011). An Essential Textual Use Case Meta-model Based on an Analysis of Existing Proposals. *14th Workshop on Requirements Engineering*, pp. 419-430.
- Li, C., van den Akker, M., Brinkkemper, S., & Diepen, G. (2010). An integrated approach for requirement selection and scheduling in software release planning. *Requirements Engineering*, 15(4), 375-396.
- Li, Z., Hall, J. G., & Rapanotti, L. (2011). On the construction of specifications from requirements. *14th Workshop on Requirements Engineering*, pp. 1-12.
- Liaskos, S., McIlraith, S.A., Sohrabi, S., & Mylopoulos, J. (2010). Integrating Preferences into Goal Models for Requirements Engineering. *18th IEEE International Requirements Engineering Conference*, pp. 135-144.
- Liegl, P., Schuster, R., Zapletal, M., Huemer, C., Werthner, H., Aigner, M., Bernauer, M., Bjoern Klinger, B., Mayr, M., Mizani, R., & Windisch, M. (2009). [vem:xi:] - A Methodology for Process Based Requirements Engineering. *17th IEEE International Requirements Engineering Conference*, pp. 193-202.
- Liu, C., Wang, Y., & Zhang, W. (2010). Elicitation of Dependability Requirements: A HAZOP-based Approach. *18th IEEE International Requirements Engineering Conference*, pp. 407-408.
- Liu, S. (2009). Integrating top-down and scenario-based methods for constructing software specifications. *Information and Software Technology*, 51(11), 1565-1572.
- Lockerbie, J., Bush, D., Maiden, Neil, Blom, H., & Everdij, M. (2010). Using i* Modelling as a Bridge between Air Traffic Management Operational Concepts and Agent-based Simulation Analysis. *18th IEEE International Requirements Engineering Conference*, pp. 351-356.

- Lohmann, S., Heim, P., & Lauenroth, K. (2008). Web-based Stakeholder Participation in Distributed Requirements Elicitation. *16th IEEE International Requirements Engineering Conference*, pp. 323-324.
- Lübke, D. & Schneider, K. (2008). Visualizing Use Case Sets as BPMN Processes. *Requirements Engineering Visualization*, pp. 21-25.
- Mäder, P., Gotel, O., & Philippow, I. (2008). Rule-Based Maintenance of Post-Requirements Traceability Relations. *16th IEEE International Requirements Engineering Conference*, pp. 23-32.
- Mahaux, M., Heymans, P., & Maiden, N. (2010). Making It all Up: Getting in on the Act to Improve Creative Requirements. *18th IEEE International Requirements Engineering Conference*, pp. 375-376.
- Maiden, N. (2009). Card Sorts to Acquire Requirements. *IEEE Software*, 26(3), 85-86.
- Maiden, N., Jones, S., Karlsen, K., Neill, R., Zachos, K., & Milne, A. (2010). Requirements Engineering as Creative Problem Solving: A Research Agenda for Idea Finding. *18th IEEE International Requirements Engineering Conference*, pp. 57-66.
- Mashkoo, A. & Jacquot, J.-P. (2011). Utilizing Event-B for domain engineering: a critical analysis. *Requirements Engineering*, 16(3), 191-207.
- Massey, A.K., Otto, P.N., Hayward, L.J., & Antón, A.I. (2009). Evaluating existing security and privacy requirements for legal compliance. *Requirements Engineering*, 15(1), 119-137.
- Mavin, A. & Wilkinson, P. (2010). Big Ears (The Return of “Easy Approach to Requirements Engineering”). *18th IEEE International Requirements Engineering Conference*, pp. 277-282.
- Mavin, A., Novak, M., Wilkinson, P., Maiden, N., & Lynch, P. (2008). Using Scenarios to Discover Requirements for Engine Control Systems. *16th IEEE International Requirements Engineering Conference*, pp. 235-240.
- Mavin, A., Wilkinson, P., Harwood, A., & Novak, M. (2009). Easy Approach to Requirements Syntax (EARS). *17th IEEE International Requirements Engineering Conference*, pp. 317-322.
- Maxwell, J.C. & Anton, A.I. (2009). Developing Production Rule Models to Aid in Acquiring Requirements from Legal Texts. *17th IEEE International Requirements Engineering Conference*, pp. 101-110.
- Mc Elroy, J. & Ruhe, G. (2010). When-to-release decisions for features with time-dependent value functions. *Requirements Engineering*, 15(3), 337-358.

- Menzel, I., Mueller, M., Gross, A., & Doerr, J. (2010). An Experimental Comparison Regarding the Completeness of Functional Requirements Specifications. *18th IEEE International Requirements Engineering Conference*, pp. 15-24.
- Meziane, F., Athanasakis, N., & Ananiadou, S. (2008). Generating Natural Language specifications from UML class diagrams. *Requirements Engineering*, 13(1), 1-18.
- Montabert, C., McCrickard, S.D., Winchester, W.W., & Pérez-Quiñones, M.A. (2009). An integrative approach to requirements analysis: How task models support requirements reuse in a user-centric design framework. *Interacting with Computers*, 21(4), 304-315. Elsevier B.V.
- Monteiro, M.R., Ebert, C., & Recknagel, M. (2009). Improving the Exchange of Requirements and Specifications between Business Partners. *17th IEEE International Requirements Engineering Conference*, pp. 253-260.
- Monteiro, R., Araujo, J., Amaral, V., & Patricio, P. (2010). MDgore: Towards Model-Driven and Goal-Oriented Requirements Engineering. *18th IEEE International Requirements Engineering Conference*, pp. 405-406.
- Monzon, A. (2008). A Practical Approach to Requirements Reuse in Product Families of On-Board Systems. *16th IEEE International Requirements Engineering Conference*, pp. 223-228.
- Moody, D.L., Heymans, P., & Matulevičius, R. (2009). Improving the Effectiveness of Visual Representations in Requirements Engineering: An Evaluation of i* Visual Syntax. *17th IEEE International Requirements Engineering Conference*, pp. 171-180.
- Moody, D.L., Heymans, P., & Matulevičius, R. (2010). Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation. *Requirements Engineering*, 15(2), 141-175.
- Morali, A., & Wieringa, R. (2010). Risk-based Confidentiality Requirements Specification for Outsourced IT Systems. *18th IEEE International Requirements Engineering Conference*, pp. 199-208.
- Mugridge, R. (2008). Managing agile project requirements with storytest-driven development. *IEEE Software*, 25(1), 68-75. IEEE Computer Society.
- Mukasa, K.S. & Kaindl, H. (2008). An Integration of Requirements and User Interface Specifications. *16th IEEE International Requirements Engineering Conference*, pp. 327-328.
- Mussbacher, G., Whittle, J., & Amyot, D. (2009). Semantic-Based Interaction Detection in Aspect-Oriented Scenarios. *17th IEEE International Requirements Engineering Conference*, pp. 203-212.

- Navarro, I., Leveson, N., & Lunqvist, K. (2010). Semantic decoupling: reducing the impact of requirement changes. *Requirements Engineering*, 15(4), 419-437.
- Nguyen, L. & Shanks, G. (2009). A framework for understanding creativity in requirements engineering. *Information and Software Technology*, 51(3), 655-662.
- Nissen, H.W., Schmitz, D., Jarke, M., Rose, T., Drews, P., Hesseler, F.J., & Reke, M. (2009). Evolution in Domain Model-Based Requirements Engineering for Control Systems Development. *17th IEEE International Requirements Engineering Conference*, pp. 323-328.
- Niu, N. & Easterbrook, S. (2008). Extracting and Modeling Product Line Functional Requirements. *16th IEEE International Requirements Engineering Conference*, pp. 155-164.
- Oliveira, A.R., Araújo, J., & Amaral, V. (2010). The VisualAORE DSL. *Fifth International Workshop on Requirements Engineering Visualization*, pp. 11-19.
- Omoronyia, I., Sindre, G., Roper, M., Ferguson, J., & Wood, M. (2009). Use Case to Source Code Traceability: The Developer Navigation View Point. *17th IEEE International Requirements Engineering Conference*, pp. 237-242.
- Ordinez, L., Donari, D., Santos, R., & Orozco, J. (2010). From User Requirements to Tasks Descriptions in Real-Time Systems. *13th Workshop on Requirements Engineering*, pp. 53-62.
- Patrício, L., Falcão e Cunha, J., & Fisk, R.P. (2009). Requirements engineering for multi-channel services: the SEB method and its application to a multi-channel bank. *Requirements Engineering*, 14(3), 209-227.
- Peng, X., Yu, Y., & Zhao, W. (2011). Analyzing evolution of variability in a software product line: From contexts and requirements to features. *Information and Software Technology*, 53(7), 707-721.
- Pérez, F. & Valderas, P. (2009). Allowing End-Users to Actively Participate within the Elicitation of Pervasive System Requirements through Immediate Visualization. *Fourth International Workshop on Requirements Engineering Visualization*, pp. 31-40.
- Pimentel, J., Castro, J., & Franch, X. (2011). Specification of Failure-Handling Requirements as Policy Rules on Self-Adaptive Systems. *14th Workshop on Requirements Engineering*, pp. 345-356.
- Pires, P.F., Delicato, F.C., Cóbe, R., Batista, T., Davis, J.G., & Song, J.H. (2011). Integrating ontologies, model driven, and CNL in a multi-viewed approach for requirements engineering. *Requirements Engineering*, 16(2), 133-160.

- Post, H., Sinz, C., Merz, F., Gorges, T., & Kropf, T. (2009). Linking Functional Requirements and Software Verification. *17th IEEE International Requirements Engineering Conference*, pp. 295-302.
- Qureshi, N.A. & Perini, A. (2010). Requirements Engineering for Adaptive Service Based Applications. *18th IEEE International Requirements Engineering Conference*, pp. 108-111.
- Racheva, Z., Daneva, M., Sikkil, K., Wieringa, R., & Herrmann, A. (2010). Do We Know Enough about Requirements Prioritization in Agile Projects: Insights from a Case Study. *18th IEEE International Requirements Engineering Conference*, pp. 147-156.
- Rashid, A. (2008). Aspect-Oriented Requirements Engineering: An Introduction. *16th IEEE International Requirements Engineering Conference*, pp. 306-309.
- Recio, R., Salzberg, C., Palm, J., & Machuca, C. (2008). Leveraging Collaborative Technologies in the IO Requirements Process. *16th IEEE International Requirements Engineering Conference*, pp. 283-288.
- Regev, G., Hayard, O., Gause, D. C., & Wegmann, A. (2009). Modeling Service-Level Requirements: A Constancy Perspective. *17th IEEE International Requirements Engineering Conference*, pp. 231-236.
- Regnell, B., Berntsson Svensson, R., & Olsson, T. (2008). Supporting roadmapping of quality requirements. *IEEE Software*, 25(2), 42-47.
- Rifaut, A. & Dubois, E. (2008). Using Goal-Oriented Requirements Engineering for Improving the Quality of ISO/IEC 15504 based Compliance Assessment Frameworks. *16th IEEE International Requirements Engineering Conference*, pp. 33-42.
- Rohleder, C. (2008). Visualizing the Impact of non-Functional Requirements on Variants: A Case Study. *Requirements Engineering Visualization*, pp. 11-20.
- Romero, M., Vizcaíno, A., & Piattini, M. (2008). Designing a simulator for the training of software engineers in global requirements elicitation. *11th Workshop on Requirements Engineering*, pp. 217-222.
- Rosado, D.G., Fernández-Medina, E., López, J., & Piattini, M. (2010). Analysis of Secure Mobile Grid Systems: A systematic approach. *Information and Software Technology*, 52(5), 517-536. Elsevier B.V.
- Rubin, E. & Rubin, H. (2010). Supporting agile software development through active documentation. *Requirements Engineering*, 16(2), 117-132.
- Rubin, E. & Rubin, H. (2011). Supporting agile software development through active documentation. *Requirements Engineering*, 16(2), 117-132.

- Sahraoui, A.-E.-K., Buede, D.M., & Sage, A.P. (2008). Systems engineering research. *Journal of Systems Science and Systems Engineering*, 17(3), 319-333.
- Salinesi, C., Mazo, R., Diaz, D., & Djebbi, O. (2010). Using Integer Constraint Solving in Reuse Based Requirements Engineering. *18th IEEE International Requirements Engineering Conference*, pp. 243-251.
- Salinesi, C., Rolland, C., Diaz, D., & Mazo, R. (2009). Looking for Product Line Feature Models Defects: Towards a Systematic Classification of Verification Criteria. *17th IEEE International Requirements Engineering Conference*, pp. 385-386.
- Sankar, K., & Hejmadi, A.S. (2008). Pattern Oriented Requirements towards Independent Software Vendors. *16th IEEE International Requirements Engineering Conference*, pp. 253-258.
- Santos, E.G. & Medeiros, A.P.D. (2011). Design Rationale Representation in Requirements Engineering using the KAOS meta-model. *14th Workshop on Requirements Engineering*, pp. 383-394.
- Savolain, J., Kuusela, J., & Vilavaara, A. (2010). Transition to Agile Development - Rediscovery of Important Requirements Engineering Practices. *18th IEEE International Requirements Engineering Conference*, pp. 289-294.
- Schmitz, D., Nissen, H.W., Jarke, M., Rose, T., Drews, P., Hesseler, F.J., & Reke, M. (2008). Requirements Engineering for Control Systems Development in Small and Medium-Sized Enterprises. *16th IEEE International Requirements Engineering Conference*, pp. 229-234.
- Sellier, D., Mannion, M., & Mansell, J. X. (2008). Managing requirements interdependency for software product line derivation. *Requirements Engineering*, 13(4), 299-313.
- Seresht, S. M., & Ormandjieva, O. (2008). Automated assistance for use cases elicitation from user requirements text. *11th Workshop on Requirements Engineering*, pp. 128-139.
- Seyff, N., Graf, F., & Maiden, N. (2010). Using Mobile RE Tools to Give End-Users Their Own Voice. *18th IEEE International Requirements Engineering Conference*, pp. 37-46.
- Seyff, N., Maiden, N., Karlsen, K., Lockerbie, J., Grünbacher, P., Graf, F., & Ncube, C. (2009). Exploring how to use scenarios to discover requirements. *Requirements Engineering*, 14(2), 91-111.
- Siddiqi, J., & Shekaran, M.C. (1996). Requirements Engineering: The Emerging Wisdom. *IEEE Software*, 13, 15-18.

- Siegl, S., Hielscher, K.-S., & German, R. (2010). Model Based Requirements Analysis and Testing of Automotive Systems with Timed Usage Models. *18th IEEE International Requirements Engineering Conference*, pp. 345-350.
- Silva, C., Borba, C., & Castro, J. (2011). A Goal Oriented Approach to Identify and Configure Feature Models for Software Product Lines. *14th Workshop on Requirements Engineering*, pp. 395-406.
- Sim, S.E., Alspaugh, T.A., & Al-Ani, B. (2008). Marginal Notes on Amethodical Requirements Engineering: What Experts Learned from Experience. *16th IEEE International Requirements Engineering Conference*, pp. 105-114.
- Singh, S.N. & Woo, C. (2009). Investigating business-IT alignment through multi-disciplinary goal concepts. *Requirements Engineering*, 14(3), 177-207.
- Sinha, A., Paradkar, A., Takeuchi, H., & Nakamura, T. (2010). Extending Automated Analysis of Natural Language Use Cases to Other Languages. *18th IEEE International Requirements Engineering Conference*, pp. 364-369.
- Smith, M.H. & Havelund, K. (2008). Requirements Capture with RCAT. *16th IEEE International Requirements Engineering Conference*, pp.183-192.
- Smith, R. & Gotel, O. (2008). Gameplay to Introduce and Reinforce Requirements Engineering Practices. *16th IEEE International Requirements Engineering Conference*, pp. 95-104.
- Soares, M. D. S., Vrancken, J., & Verbraeck, A. (2011). User requirements modeling and analysis of software-intensive systems. *Journal of Systems and Software*, 84(2), 328-339.
- Sommerville, I. & Sawyer, P. (2007) Requirements engineering: a good practice guide, Chichester, Wiley.
- Song, X., Hwong, B., & Ros, J. (2009). Experiences in Developing Quantifiable NFRs for the Service-Oriented Software Platform. *17th IEEE International Requirements Engineering Conference*, pp. 337-342.
- Song, X., Hwong, B., & Ros, J. (2011). Lessons from Developing Non-Functional Requirements for a Software Platform. *IEEE Software*, [doi.ieeecomputersociety.org/10.1109/MS.2011.69](https://doi.org/10.1109/MS.2011.69).
- Stoiber, R., Fricker, S., Jehle, M., & Glinz, M. (2010). Feature Unweaving: Refactoring Software Requirements Specifications into Software Product Lines. *18th IEEE International Requirements Engineering Conference*, pp. 403-404.
- Sun, L. & Mushi, C.J. (2010). Case-based analysis in user requirements modelling for knowledge construction. *Information and Software Technology*, 52(7), 770-777.

- Sundaram, S.K., Hayes, J.H., Dekhtyar, A., & Holbrook, E.A. (2010). Assessing traceability of software engineering artifacts. *Requirements Engineering*, 15(3), 313-335.
- Supakkul, S. & Chung, L. (2010). Visualizing non-functional requirements patterns. *Fifth International Workshop on Requirements Engineering Visualization*, pp. 25–34.
- Supakkul, S., Hill, T., Chung, L., Tun, T.T., & Leite, J.C.S.P. (2010). An NFR Pattern Approach to Dealing with NFRs. *18th IEEE International Requirements Engineering Conference*, pp. 179-188.
- Sustar, H., Pfeil, U., & Zaphiris, P. (2008). Requirements elicitation with and for older adults. *IEEE Software*, 25(3), 16–17.
- Sutcliffe, A., Thew, S., & Jarvis, P. (2011). Experience with user-centred requirements engineering. *Requirements Engineering*.
- Sutherland, M. & Maiden, N. (2010). Storyboarding Requirements. *IEEE Software*, 27(December), 9-11.
- Svahnberg, M., Gorschek, T., Feldt, R., Torkar, R., Saleem, S.B., & Shafique, M.U. (2010). A systematic review on strategic release planning models. *Information and Software Technology*, 52(3), 237-248.
- Tanabe, D., Uno, K., Akemine, K., Yoshikawa, T., Kaiya, H., & Saeki, M. (2008). Supporting Requirements Change Management in Goal Oriented Analysis. *16th IEEE International Requirements Engineering Conference*, pp. 3-12). Barcelona.
- Thew, S. & Sutcliffe, A. (2008). Investigating the Role of “Soft Issues” in the RE Process. *16th IEEE International Requirements Engineering Conference*, pp. 63-66.
- Thew, S., Sutcliffe, A., Bruijn, O.D., McNaught, J., Procter, R., Venters, C., & Buchan, I. (2008). Experience in e-Science Requirements Engineering. *16th IEEE International Requirements Engineering Conference*, pp. 277-282.
- Tjong, S.F. & Berry, D.M. (2008). Can Rules of Inferences Resolve Coordination Ambiguity in Natural Language Requirements Specification? *11th Workshop on Requirements Engineering*, pp. 205-210.
- Tøndel, I.A., Jaatun, M.G., & Meland, P.H. (2008). Security requirements for the rest of us: A survey. *IEEE Software*, 25(1), 20–27.
- Trujillo, J., Soler, E., Fernández-Medina, E., & Piattini, M. (2009). An engineering process for developing Secure Data Warehouses. *Information and Software Technology*, 51(6), 1033-1051.

- Tun, T.T., Jackson, M., Laney, R., Nuseibeh, B., & Yu, Y. (2009). Are Your Lights Off? Using Problem Frames to Diagnose System Failures. *17th IEEE International Requirements Engineering Conference*, pp. 343-348.
- Valderas, P., & Pelechano, V. (2009). Introducing requirements traceability support in model-driven development of web applications. *Information and Software Technology*, 51(4), 749-768.
- Van Lamsweerde, A. (2001) Goal-Oriented Requirements Engineering: A Guided Tour. *Fifth IEEE International Symposium on Requirements Engineering*, p. 249
- Villela, K., Doerr, J., & Gross, A. (2008). Proactively Managing the Evolution of Embedded System Requirements. *16th IEEE International Requirements Engineering Conference*, pp. 13-22.
- Vivas, J.L., Agudo, I., & López, J. (2010). A methodology for security assurance-driven system development. *Requirements Engineering*, 16(1), 55-73.
- Waldmann, B. & Jones, P. (2009). Feature-oriented Requirements Satisfy Needs for Reuse and Systems View. *17th IEEE International Requirements Engineering Conference*, pp. 329-334.
- Wang, B., Zhang, W., Zhao, H., Jin, Z., & Mei, H. (2009). A Use Case Based Approach to Feature Models' Construction. *17th IEEE International Requirements Engineering Conference*, pp. 121-130.
- Weber-Jahnke, J.H. & Onabajo, A. (2009). Finding Defects in Natural Language Confidentiality Requirements. *17th IEEE International Requirements Engineering Conference*, pp. 213-222.
- Weiss, M. & Mouratidis, H. (2008). Selecting Security Patterns that Fulfill Security Requirements. *16th IEEE International Requirements Engineering Conference*, pp. 169-172.
- Wen, Y., Zhang, H., Liu, L., & Yang, H. (2010). One bridge, two gaps-beyond an engineering approach: Creativity in requirements elicitation. *Fifth International Workshop on Requirements Engineering Visualization*, pp. 40-42.
- Werneck, V.M.B., Oliveira, A.P.A., & do Leite, J.C.S.P. (2009). Comparing GORE Frameworks: i-star and KAOS. *12th Workshop on Requirements Engineering*, pp. 15-26.
- Weston, N., Chitchyan, R., & Rashid, A. (2008). A Formal Approach to Semantic Composition of Aspect-Oriented Requirements. *16th IEEE International Requirements Engineering Conference*, pp. 173-182.
- Weston, N., Chitchyan, R., & Rashid, A. (2009). Formal semantic conflict detection in aspect-oriented requirements. *Requirements Engineering*, 14(4), 247-268.

- Whittle, J., Sawyer, P., Bencomo, N., Cheng, B.H.C., & Bruel, J.-M. (2009). RELAX: Incorporating Uncertainty into the Specification of Self-Adaptive Systems. *17th IEEE International Requirements Engineering Conference*, pp. 79-88.
- Whittle, J., Sawyer, P., Bencomo, N., Cheng, B.H.C., & Bruel, J.-M. (2010). RELAX: a language to address uncertainty in self-adaptive systems requirement. *Requirements Engineering*, 15(2), 177-196.
- Winkler, S. (2008). On Usability in Requirements Trace Visualizations. *Requirements Engineering Visualization*, pp. 56-60.
- Wnuk, K., Callele, D., & Regnell, B. (2010). Guiding Requirements Scoping Using ROI: Towards Agility, Openness and Waste Reduction. *18th IEEE International Requirements Engineering Conference*, pp. 409-410.
- Wnuk, K., Regnell, B., & Karlsson, L. (2008). Visualization of Feature Survival in Platform-Based Embedded Systems Development for Improved Understanding of Scope Dynamics. *Requirements Engineering Visualization*, pp. 41-50.
- Wnuk, K., Regnell, B., & Karlsson, L. (2009). What Happened to Our Features? Visualization and Understanding of Scope Change Dynamics in a Large-Scale Industrial Setting. *17th IEEE International Requirements Engineering Conference*, pp. 89-98.
- Wolter, C. & Meinel, C. (2010). An approach to capture authorisation requirements in business processes. *Requirements Engineering*, 15(4), 359-373.
- Wolter, K., Smialek, M., Bildhauer, D., & Kaindl, H. (2008). Reusing Terminology for Requirements Specifications from WordNet. *16th IEEE International Requirements Engineering Conference*, pp. 325-326.
- Yang, H., Roeck, A., Gervasi, V., Willis, A., & Nuseibeh, B. (2011). Analysing anaphoric ambiguity in natural language requirements. *Requirements Engineering*, DOI 10.1007/s00766-011-0119-y
- Young, J.D. (2011). Commitment analysis to operationalize software requirements from privacy policies. *Requirements Engineering*, 16(1), 33-46.
- Zachos, K., & Maiden, N. (2008). Inventing Requirements from Software: An Empirical Investigation with Web Services. *16th IEEE International Requirements Engineering Conference*, pp. 145-154.
- Zhang, Y., Harman, M.S., Finkelstein, A., & Afshin Mansouri, S. (2011). Comparing the performance of metaheuristics for the analysis of multi-stakeholder tradeoffs in requirements optimisation. *Information and Software Technology*, 53(7), 761-773.