Hand in H5

Software Architecture in Practice

Architecture design methods

Architecture design decisions

Department of Computer Science, University of Aarhus

Aabogade 34, 8200 Århus N, Denmark

Gruppe: Bravo

20074842, Lars Kringelbach, lars@kringelbach.com

20064684, Marjus Nielsen, Marjus.nielsen@gmail.com

20074877, Morten Herman Langkjær, morten.herman@gmail.com

20054680, Peter Madsen, pm@chora.dk

<<Dato: 12. maj 2008>>

1 Introduktion 3
2 H5a - Describing an Architecture Design Method 3
2.1 HS-07 organisation 3
2.2 Tailored architecture design method 4
2.2.1 Architectural analysis 4
2.2.2 Architectural synthesis 5
2.2.3 Architectural evaluation 6
2.2.4 Overall process drivers 6
3 H5b - Documenting Design Decisions 7
3.1 Scheme for documenting design decisions 7
3.2 Rationale for decision template. 8
3.3 Template 9
4 H5c - Applying the Architecture Design Method 10
4.1 Architectural analysis 10
4.2 Architectural synthesis 10
4.2.1 Allocation viewpoint 11
4.2.2 Module viewpoint 12
4.2.3 Component & Connector viewpoints 13
4.3 Architectural Evaluation 15
4.3.1 CBAM udført på brugergrænseflade taktikker 15
4.3.2 aSQA 21
4.4 Overall process driver 22
5 H5d - Summary and Discussion 23
6 References 24

1 Introduktion

This exercise focuses on architectural design method and architectural design decisions. You will describe a tailored architecture design method (in H5a), apply it to a new version of HS-07 (HS-08, in H5b), and finally practice documenting architectural design decisions (in H5c).

2 H5a - Describing an Architecture Design Method

"Consider the grid of Table 4 in Hofmeister et al. [1, p. 121]. In this section, you should use the knowledge about software architecture to decide on artifacts, activities, and techniques and tools for a tailored architecture design method. The method should fit the organization producing HS-07[1] and you should use it in the next exercise. Describe your choices and argue for why you made them."

2.1 HS-07 organisation

En arkitektur designmetode skal tilpasses den enkelte organisation og den type produkter der fremstilles. En organisation på fem personer kan kun vanskeligt gennemføre en fuld ATAM uden eksterne konsulenter, mens der er ressourcer til dette i større organisationer. Behovet for "tunge" reviews af arkitekturen er større hvis der fx. skal laves et system til kontrol af kraftværker end hvis der skal laves et system som ikke er mission critical, f.eks. en WEB butik.

Organisationen som fremstiller HS-07 er ukendt, hvorfor der til denne besvarelse er antaget følgende:

- Firmaets primære arbejdsområde er at lave Embedded software.
- Virksomhedens størrelse er ca. 200 mand
- Heraf er der et team med ansvar for udvikling af HS-07:
    - En produkt-ejer
    - En arkitekt
    - En sælger
    - Fem udviklere

Der er med andre ord tale om et mellemstort firma med et forholdsvis lille team med ansvar for HS-07. Da teamet er relativt lille er der behov for letvægtsmetoder, mens det til gengæld må forventes at folk fra andre dele af organisationen kan være tilgængelige i forbindelse med reviews mv.

Gruppen har vurderet at HS-07 er et produkt som udvikles internt, i modsætning til produkter som udvikles til en specifik kunde. Der findes derfor ikke nogen kunde til systemet i gængs forstand, hvorfor produkt-ejer tænkes at udfylde rollen som kunde. Sælger har også indflydelse på kravbilledet, da denne forsøger at

[1] Speculate on characteristics of the organization as needed

agere på et marked bl.a. ved at få nye features med i systemet så det dækker specifikke større kunders krav.

2.2 Tailored architecture design method

Der er behov for en arkitektur design metode som er tilpasset organisationen beskrevet ovenfor. Med udgangspunkt i [1] er gruppen nået frem til en model som følger (se nedenfor for beskrivelse og argumentation):

	Generic artifact	Artifacts	Activities	Techniques and tools	Beslutningsrationale
Architectural analysis	-Context -Requirements -Architectural significant requirements	Main use cases Main quality attributes	Derive Main use cases from target market analysis QAW	-UML -QAS -Use cases	Produktkunde ikke kendt direkte
Architectural synthesis	Candidate architectural solutions – Architectural design (e.g., views, perspectives) – or Prototypes – Rationale	Viewpoints Prototype(s).	ADD Architectural prototyping	UML Skelton implementation	Letvægtsprocess
Architectural evaluation	– Quality attributes – Architectural assessment	Utility graphs Return On Investment System Health System Focus Points Prototype comments	Perform CBAM Perform aSQA Evaluate prototypes	ASQA,CBAM	Letvægtsprocess, lille projektorganistation
Overall process driver	Backlog	Backlog	Keep backlog updated	Wiki	Letvægtsprocess, kun en arkitekt
other					

2.2.1 Architectural analysis

Som tidligere beskrevet tænkes HS-07 organisationen at udvikle produkter uden ekstern kunde. Udgangspunktet for den arkitektoniske analyse tænkes derfor at være markedsanalyser, arkitektens erfaringer og forskellige kvalitetskrav fra andre dele af udviklingsorganisationen.

2.2.1.1 Artefakter

Hoved use cases:

Disse beskriver systemets hovedfunktionalitet. Disse tænkes udført på almindelig vis i klartekst. Da beskrivelserne tænkes på et relativt højt abstraktionsniveau, vil der typisk ikke være behov for yderligere beskrivelse. Evt. kan UML sekvensdiagrammer benyttes til at beskrive specielle detaljer om nødvendigt.

Kvalitetsatributter:

Liste af kvalitets-atributter for systemet. Disse beskrives med quality attribute scenarios (QAS).

2.2.1.2 Aktiviteter

Der udføres to hovedaktiviteter i denne fase: 1. beskrivelse af systemets hoved use cases og 2. gennemførsel af en quality attribute workshop (QAW).

Beskrivelse af hoved use cases:

I den beskrevne organisation tænkes det ønskede system beskrevet i form af en markedsanalyse

kombineret med kommentarer fra sælger og andre dele af organisationen. Opgaven for arkitekten er derfor at sikre sig at der udledes retvisende hoved use cases, som præcist beskriver den ønskede funktionalitet.

Quality attribute workshop:
For at få afdækket vigtige kvalitetsatributter og samtidigt sikre at relevante interessenter høres, afholdes en Quality Attribute Workshop. Det tænkes at i den beskrevne organisation deltager hele udviklingsteamet i dette arbejde.

Rationale:
Andre designmetoder producerer flere artefakter end de to ovenstående (risici, tekniske issues mv.). Det gennemgående tema i denne metode er at den skal være letvægts, for at passe til organisationen og de relativt simple produkter den producerer. Det er gruppens vurdering at der til enhver arkitekturanalyse som minimum skal reflekteres over hovedfunktionaliteten (main use cases) og de ønskede kvalitetskrav (performance, security mv.), hvorfor der udføres netop disse to aktiviteter.

2.2.2 Architectural synthesis
Da organisationen har brug for en letvægts metode og der tænkes benyttet en iterativ udviklingsproces, baseres syntesen på Attribute-Driven Design (ADD) og prototyper.

2.2.2.1 Artefakter
Viewpoints:
Output fra ADD er forskellige viewpoints. Der benyttes de tre views som beskrevet i [4]. Andre dokumentationsmetoder er mere omfattende, men det er gruppens overbevisning at ovenstående er et passende dokumentationsniveau, produktet og organisationen taget i betragtning. Historikken på de forskellige beslutninger, som tages under udførelsen af ADD, noteres i projektets backlog.

Prototyper:
Da HS-07 tænkes at være en ny type produkt for organisationen, skal der benyttes en del teknologier som gruppen ikke er fuldt fortrolige med. Af den grund er arkitekturprototyper vigtige for organistationen.

2.2.2.2 Aktiviteter
ADD:
Attribute-driven design som beskrevet i [5] er en iterativ dekomponerings proces som tager udgangspunkt i systemets krav og kvalitets attributter og som producerer en arkitektur som et antal views.

Som en del af ADD er det arkitektens opgave at vælge forskellige arkitekturer og patterns. I organisationen tænkes arkitekten her at gøre brug af egen viden og erfaringer. Det er givetvis også mulighed for at udveksle erfaringer med arkitekter i resten af organisationen, i tråd med Architecture Business Cycle som beskrevet i [5].

ADD udføres af arkitekten og den resulterende arkitektur evalueres herefter, før den og en eller flere prototyper benyttes af udviklerne.

Arkitekturprototyper:
Som tidligere skrevet anses udviklingen af prototyper som en vigtig del af arkitektens opgaver i denne organisation. Prototyperne tænkes at tjene to formål. Først og fremmest skal de give sikkerhed for at

designbeslutninger fungerer i praksis, men de tænkes også benyttet som skeletapplikation til den videre udvikling. Eksistensen af en skeletapplikation giver en vis sikkerhed for der udvikles i tråd med arkitekturen og giver et mindre behov for anden dokumentation.

2.2.3 Architectural evaluation

Der er igen fokus på at der findes en letvægtsproces til evaluering af arkitekturen. Da der udføres Quality Attribute Scenarios i forvejen er det oplagt at benytte disse som input til CBAM og aSQA.

2.2.3.1 Artefakter

Return On Investment:
Output fra CBAM. Beskriver hvilke tiltag som bedst kan betale sig at udføre.

System health, system focus points:
Output fra aSQA. Viser hvilke problemer i arkitekturen der bør fokuseres på.

Prototypeevaluering:
Resultat af evaluering af prototype. Dette benyttes til fx. at forkaste / acceptere en arkitektur ud fra om forskellige kvalitetskrav kan holdes.

2.2.3.2 Aktiviteter

CBAM:
Da der ikke udføres ATAM er der lidt arbejde med at udvælge scenarier, men H4 viste at aktiviteten giver god mening for selv små systemer. Da udviklingsgruppen er af begrænset størrelse synes det sundt at koncentrere sig om at arbejde med det som giver størst Return On Investment.

aSQA:
aSQA er en metode som arbejder på komponentniveau og er dermed også velegnet til små systemer som dette. Da den samtidigt giver værdifuld information om systemets sundhedstilstand er den oplagt at udføre løbende.

2.2.4 Overall process drivers

Backlog:
Da der kun er en arkitekt tilknyttet tænkes det at denne fører en backlog. Dette dog på elektronisk form, så det er muligt for andre at tilgå informationen.

3 H5b - Documenting Design Decisions

"Consider the templates for describing design decisions in [2] and [3]. Describe a scheme for documenting design decisions that would fit the organization producing HS-07.

Describe your choices and argue for why you made them."

3.1 Scheme for documenting design decisions

Følgende information skal beskrives for de beslutninger der tages under design af arkitekturen for HS08:

Kravene er udledt ud fra en organisation som dokumenteret i afsnit 2.1. Rationalet for strukturen beskrives i afsnit 3.2.

Problembeskrivelse:

- Der skal være en beskrivelse af det problem beslutningen omhandler. Denne beskrivelse er nødvendig for fx at kunne identificere om der er flere beslutninger der omhandler det samme problem. Dette er specielt vigtigt for at sikre at der ikke tages modstridende beslutninger. Problembeskrivelsen hjælper også med til at identificere hvornår man har løst problemet. Det er problembeskrivelsen der identificerer beslutningen.

Beskrivelse af beslutning:

- Der skal være en beskrivelse af den valgte beslutning, som løser det beskrevne problem.

Beslutningens karakter:

- Dette felt beskriver hvilke kvaliteter beslutningen har indflydelse på (fx modifiability, performance), samt hvilken effekt indflydelsen har (fx øget performance). Denne information kan bruges til at identificere og udfordre beslutninger på et senere tidspunkt, hvis det fx viser sig at nogle kvalitetskrav ikke er opfyldt.

Rationale:

- Rationalet begrunder valget af beslutning, både med grunde for den valgte løsning samt grunde for at alternativerne er valgt fra. Det kan være en vurdering ud fra en struktureret analyse af problemet, fx Structured Decision Making (SDM), udførte tests, undersøgelser, arkitekt-erfaring (fra andre systemer eller litteratur) etc.

- Rationalet må også gerne indeholde relationer til andre beslutninger, artefakter eller krav m.v.

Antagelser:

- Hvis der er lavet nogle antagelser i valget af løsning skal de beskrives. Det er nødvendigt at have antagelserne beskrevet, så man kan omgøre beslutningen på et senere tidspunkt, hvis det viser sig at de ikke er korrekte. Dette gælder både antagelser om den valgte løsning, samt evt. antagelser om alternativerne.

Implikationer:

- Hvis der er nogen implikationer ved den valgte løsning skal de beskrives. Det kan fx være at den valgte løsning giver anledning til nogle problemer andre steder i systemet, eller at den giver nogle begrænsninger på andre beslutninger.

Alternativer:

- Alternativer til den valgte løsning skal beskrives kort, så man kan se hvilke muligheder der er blevet overvejet. Det kan bekræfte at beslutningen er taget ud fra et fornuftigt grundlag, eller give anledning til yderligere undersøgelser hvis der opstår flere muligheder.

Status (Nuværende status, historik for ændringer, tidsstempler):

- Beslutningens nuværende status skal beskrives (afventer, godkendt, afvist) samt et tidsstempel for hvornår den sidst er ændret. Der skal desuden være en historik af de ændringer der er blevet foretaget mht. denne beslutning, fx hvis man på et tidspunkt har valgt et andet alternativ. Denne historik sikrer at en beslutning kan omgøres på en styret måde, så man ikke vender tilbage til tidligere afviste løsninger. Alle ændringer skal have et tidsstempel for ændringen.

3.2 Rationale for decision template.

Rationalet for udformningen af decision templaten ligger bl.a. i den type organisation som vi forventer skal udføre dokumentationen af beslutningerne, og dermed også bruge dem i fremtiden. På trods af at organisationen i sig selv må betragtes som mellemstor, så er den udøvende del af organisationen relativt lille. Det betyder også at der må forsøges at finde en balance imellem et tilpas pragmatisk niveau af dokumentation, samt en tilpas omfattende dokumentation til at den skaber den nødvendige værdi.

Problembeskrivelse og beslutningsbeskrivelse er absolutte nødvendigheder for at kunne træffe en beslutning.

Beslutningens karakter er et karakteristika ved en beslutning, som kan udelades, men da netop karakteren af beslutningen er med til at dokumentere hvilke kvaliteter der påvirkes, er denne information nødvendig i beslutninger som relaterer sig til arkitekturarbejde.

Rationalet for beslutningen er det input som kvalificerer hvorfor beslutningen er faldet ud som den er, og dermed en nødvendighed hvis man skal have gavn af beslutningen i et længere perspektiv. Dette gør sig gældende for vores organisation da det er et produkt der udvikles, og dermed må forventes at have en lang levetid. Relationer til andre beslutninger er ikke et must, men kan bidrage med værdifuld information til at spore og krydsreferere udviklingen af beslutninger over tid. Vi mener kun det bør dokumenteres i det omfang det er meget værdiskabende, i andre tilfælde kan denne reference udelades.

Antagelser angiver de basisfordusætninger der ligger til grund for beslutningen, og da disse kan forandre sig over tid, er det en nødvendiighed at dokumentere disse.

Statusinformation er en nødvendighed, for at kunne se om beslutningen er accepteret, ændret, omgjort eller andet. Det er nødvendigt for i et historisk perspektiv at kunne spore hvor vidt en beslutning forbliver

gældende, samt eventuelt om den er omgjort, og alternativer er afprøvet istedet. Derfor indeholder status også en historik over beslutningen.

3.3 Template

På baggrund af de foregående afsnit, er der udviklet følgende decision template som vil blive brugt til at dokumentere beslutninger for HS08:

Information		Dokumentation
Problembekrivelse		
Beskrivelse af beslutning		
Beskrivelsens karakter		
Rationale	SDM eller andet rationale	
	Relationer til andre beslutninger, artefakter, krav mm.	
Antagelser		
Implikationer		
Alternativer		
Status	Nuværende status	
	Historik for ændring af beslutning, evt. henvisninger til alternativer	

Skraveringsbetydning:	Obligatorisk udfyldning	Optionel udfyldning

4 H5c - Applying the Architecture Design Method

"Here, you must apply the design method you described in H5a on the HS-08 system which extends HS-07. The following is the vision of HS-08:

HS-08 is to be an evolutionary extension of HS-07 in response to new stakeholder requirements following the success of HS-07. In particular, the following areas should be designed:

1. **Web access to HS-08. HS-08 needs a modern and user friendly web access that is secure for both house occupants and potential house management**
2. **New sensors and actuators for HS-08. HS-08 must support new sensors and actuators. In particular it has been a consistent requirement that security equipment (movement sensors, video cameras, and sound alarms) should be integrated**

Your task is in principle to design a revision of HS-07 that takes both of these areas into consideration. However, we are not concerned with coming up with a full solution here, but rather with taking several small design iterations. This means that you do not have to do full requirement elicitation, view description, evaluation etc.

During the exercise, you must use the SAiP Wiki

http://wiki.daimi.au.dk/saip2008

to

1. **Maintain a backlog (as described in [1])**
2. **Document your design decisions (according to H5b)**

You must use the Wiki page for your specific group. Your user name is the name of your group and the password is "saip2008". Please remember to write your name at the bottom of the page when you edit it."

4.1 Architectural analysis

Arkitektur-analysen er foregået direkte på wikien og der henvises hertil for main use cases og kvalitetsattributscenarier[2].

4.2 Architectural synthesis

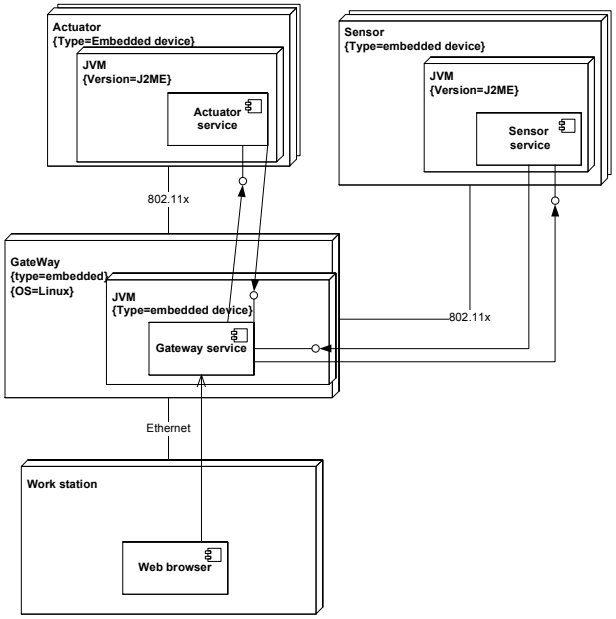
I den arkitektoniske design metode, er der truffet det valg at den arkitektoniske syntese består af viewpoints fra [5]. Yderligere supporteres disse views med en eller flere prototype implementationer for at sandsynliggøre at arkitekturen kan opfylde de fastlagte kvaliteter. Da der ikke er lagt op til at det er vigtigt at lave en fuldt dækkende udførsel af vores designmetode, er prototypen fravalgt i opgaven, og der er lavet

[2] http://wiki.daimi.au.dk/saip2008/bravo.wiki?cmd=get&anchor=Bravo

viewpoints som illustrerer hvordan nye typer hardware kan integreres i systemet. Der er lavet et antal C&C view points som illustrerer kørselsopførsel af gateway, samt registrering af nye sensorer og aktuatorer og indbyrdes ansvarsområder. Yderligere er der lavet et allocation viewpoint, som illustrerer hvordan omverdenen kan kontrolere systemet via en webbrowser. Module viewpoint er identisk med det oprindelige system, og er indsat af illustrative hensyn.

4.2.1 Allocation viewpoint

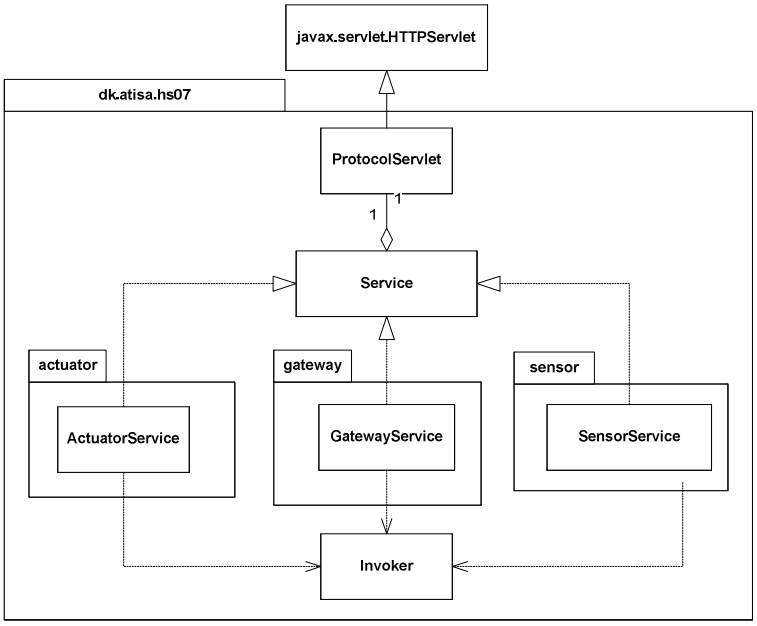
På allocation viewpoint, kan det ses at alle de eksisterende komponenter eksisterer, hvor radiator og termometer dog er erstattet af henholdsvis actuator og sensor, for at generalisere hardware typer. Yderligere er tilføjet Web-browser komponent på en workstation, som illustrerer udvidelsen af systemet med henblik på integration fra en ekstern node, via en browser og ethernet. Web-browser integrationen er kun til stede på allocation viewpoint, da der i component & connector viewpoint er fokuseret på udvidelse af systemet med nye hardware typer.



Figur 4-1 Allocation viewpoint

4.2.2 Module viewpoint

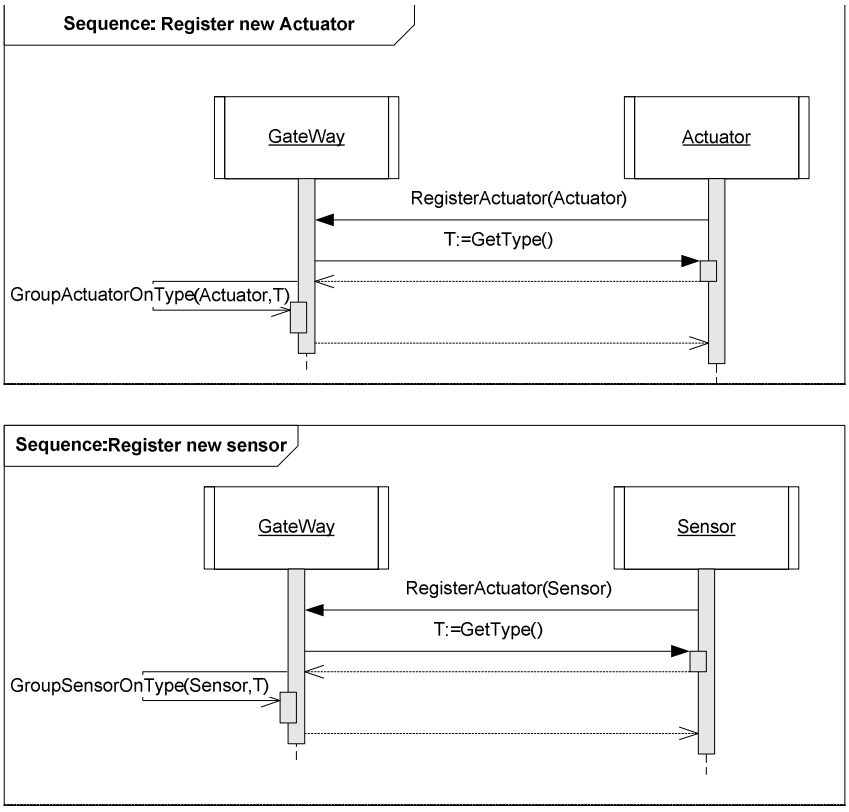
Module viewpoint er identisk med den eksisterende nedbrydning, dog er der igen ændret ved navngivningen af services, således at RadiatorService og ThermometerService er omdøbt til ActuatorService og SensorService.



Figur 4-2 Module viewpoint

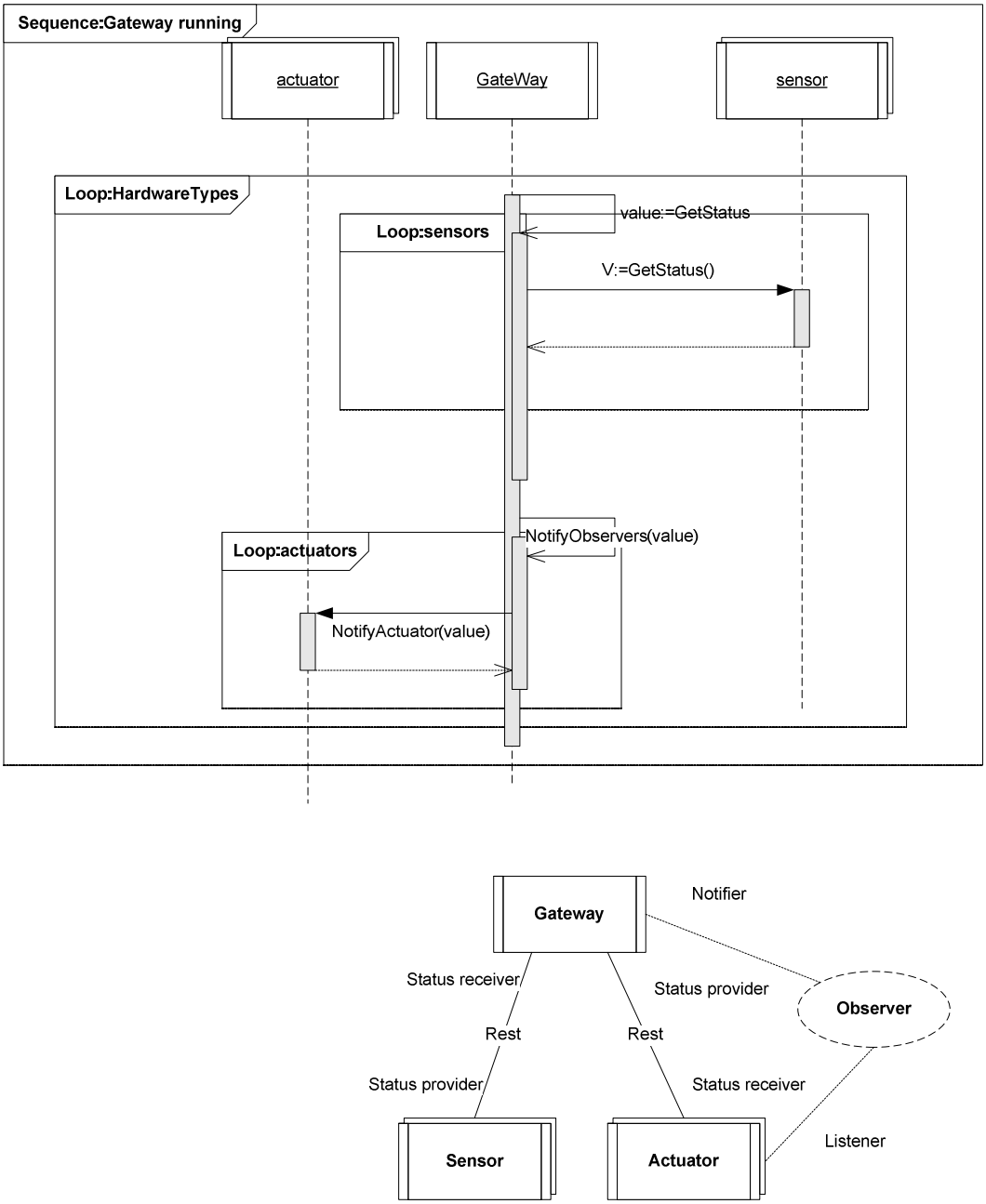
4.2.3 Component & Connector viewpoints

I component & connector viewpoint er der ændret i sekvensen hvormed en Actuator eller Sensor registrerer sig på Gateway. Hvor der tidligere kun blev overført en adresse hvorpå Gateway kan kontakte henholdsvis Actuator og Sensor, så tilføjes der en metode på hardwaren som hedder GetType, og som kan benyttes til at få information om hardware-typen som forsøger at forbinde til en gatewayen. På den måde er det muligt at gruppere aktuatorer og sensorer efter hardware-type, og dermed sikre at information fra en type sensor når den tilsvarende aktuator.



Figur 4-3 Sekvens registrer ny aktuator og sensor

På Figur 4-4 kan det ses at når gatewayen kommunikerer med sensor og aktuator, grupperes disse efter hardware-type, Der itereres over sensorer indenfor en given hardware-type, og de læsninger som modtages distribueres videre til aktuatorer af samme type. Af Figur 4-4 ses også at der er en ændring i ansvarsfordelingen imellem komponenterne, hvor der tidligere var en temperatur provider og temperatur receiver, så er det nu ændret til en status provider og status receiver. Begge ansvar er mere generaliserede en tidligere, og underbygger den højere grad af modifiability som er påkrævet for at kunne udvide systemet med nye hardware-typer under kørsel af
systemet.



Figur 4-4 Sekvens for kørende gateway samt C&C for ansvar og indbyrdes forhold

4.3 Architectural Evaluation

Som evaluering af arkitekturen er der lavet en CBAM undersøgelse af de forskellige muligheder der er for at tilføje en brugergrænseflade til systemet. I analysen overvejes der både om en webbaseret løsning er den rigtige samt hvor stor fleksibilitet og sikkerhed der skal være i den.

4.3.1 CBAM udført på brugergrænseflade taktikker

4.3.1.1 Udvælgelse af scenarier

Der er lavet en CBAM undersøgelse for at vurdere hvilke taktikker vi skal bruge for at tilføje en bedre brugergrænseflade til HS-08-systemet.

1. **A user accesses the system using a web based user interface.**

   Systemet skal have et brugerinterface så det er muligt at tilgå systemet på en nem og brugervenlig måde. Målet er at systemet skal kunne benyttes uden brug af manualer. Da funktionaliteterne i brugergrænsefladen er relativt enkle vil en normal web brugergrænseflade være tilstrækkelig. Hvis der skal designes en ekstra brugervenlig version, vil der være yderligere omkostninger under udviklingen.

2. **An administrator configures the system using a web based user interface.**

   Der skal være en gruppering af brugere, så det kun er udvalgte brugere der kan konfigurere gatewayen mht. brugere, IP-adresser, nye sensorer og andre statiske indstillinger. Almindelige brugere skal kun kunne checke status samt ændre på systemets tilstand som fx ønsket temperatur, se videosignaler eller til-/frakobling af alarm.

3. **A user tries to modify the settings of the system via the web interface, the system checks the authenticity of the user and only performs the operation if the user is authorized.**

   Det skal sikres at det kun er brugere der er kendt af systemet der kan få adgang til at udføre operationer på gatewayen. Målet er at der er sikker adgang til følsomme informationer, hvilket både gælder styring af alarmen og konfiguration af gatewayen.

4.3.1.2 Respons-mål

I følgende tabel er respons-målene summeret med værste, nuværende, ønskede og bedste niveau angivet:

Response goals				
Scenario	**Worst**	**Current**	**Desired**	**Best**
1	Ingen UI	Ingen UI	Web UI	Let Web UI
2	Ingen gruppering	Ingen gruppering	2 grupper	Fuldt konfigurerbar
3	Ingen sikkerhed	Ingen sikkerhed	Høj sikkerhed (SSL)	Høj sikkerhed (SSL)

4.3.1.3 Prioriterring

4 stakeholders har tildelt 100 stemmer hver til de 3 scenarier:

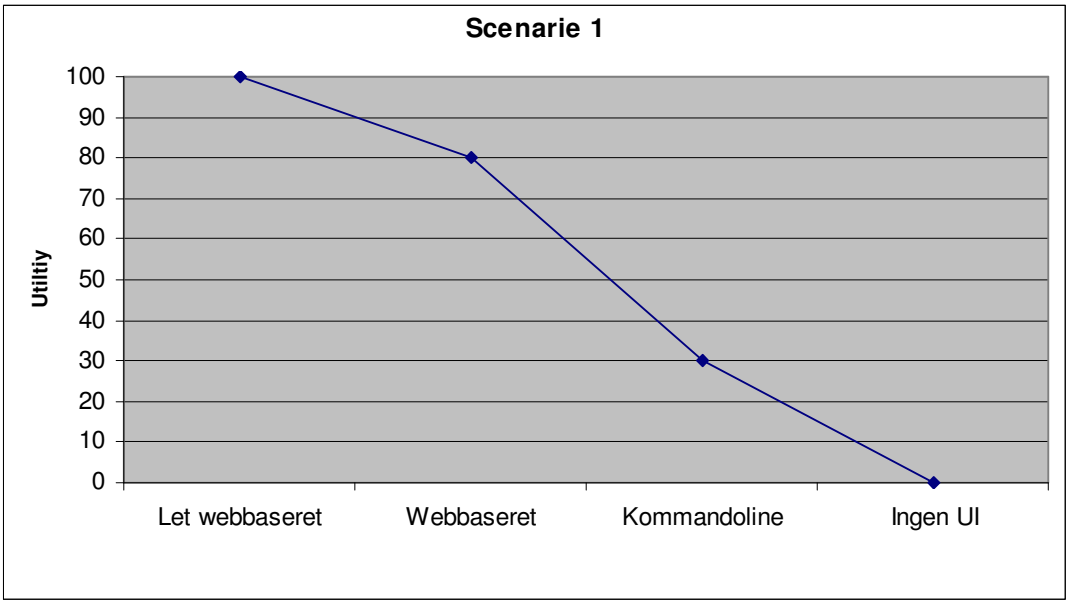
Response goals - Prioritization					
Scenario	**Votes**	**Worst**	**Current**	**Desired**	**Best**
1	175	Ingen UI	Ingen UI	Web UI	Web UI
2	50	Ingen gruppering	Ingen gruppering	2 grupper	Fuldt konfigurerbar
3	175	Ingen sikkerhed	Ingen sikkerhed	Høj sikkerhed (SSL)	Høj sikkerhed (SSL)

Det er vurderet at det meget vigtigt at få en brugervenlig brugergrænseflade med meget høj sikkerhed. Det er mindre vigtigt at der er en opdeling af brugere i grupper.

4.3.1.4 Tildel værdi

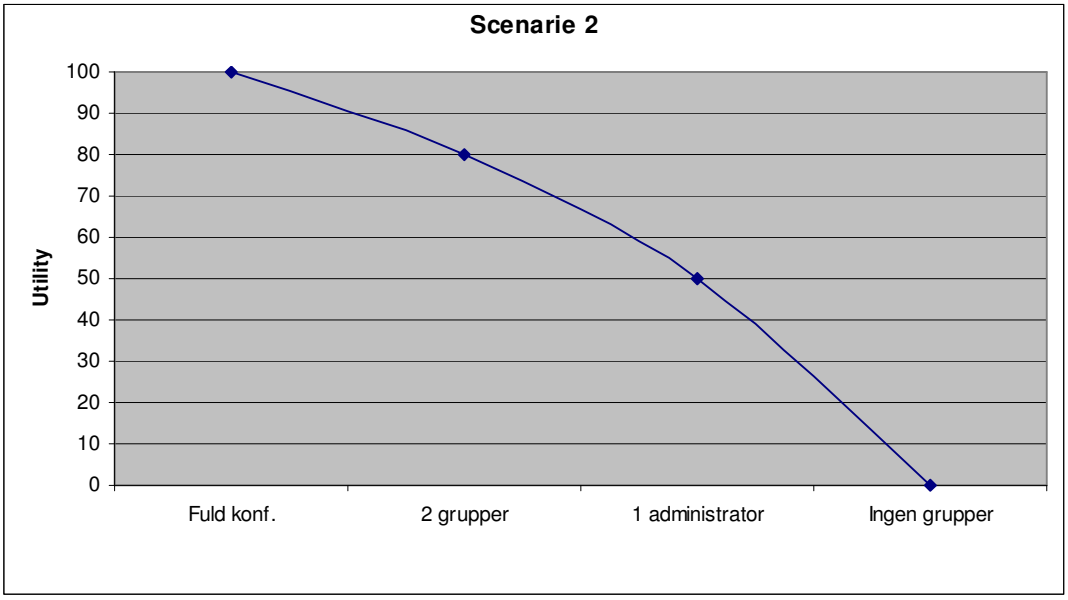
Følgende Utility grafer viser hvor meget værdi et respons har i forhold til hvor godt responsmålet opfyldes.

4 muligheder er blevet sammenlignet i analysen af scenarie 1. Der er ikke nogen brugergrænseflade i systemet, hvilket er uacceptabelt. En kommandolinie brugergrænseflade vil kunne give brugeren mulighed for at foretage de nødvendige operationer, men det vil ikke være brugervenligt og let tilgængeligt fra fx. Internettet. En webbaseret brugergrænseflade kan relativt nemt laves så den kan bruges uden brug af manualer da det hovedsageligt er relativt enkle operationer der skal foretages. Man kan vælge at få designet en meget brugervenlig brugergrænseflade med hjælpefunktioner og muligheder for at lave makrofunktioner der kan sammensætte ofte brugte funktioner (aggregerede funktioner), men dette vil give ekstra omkostninger både under design og udvikling.

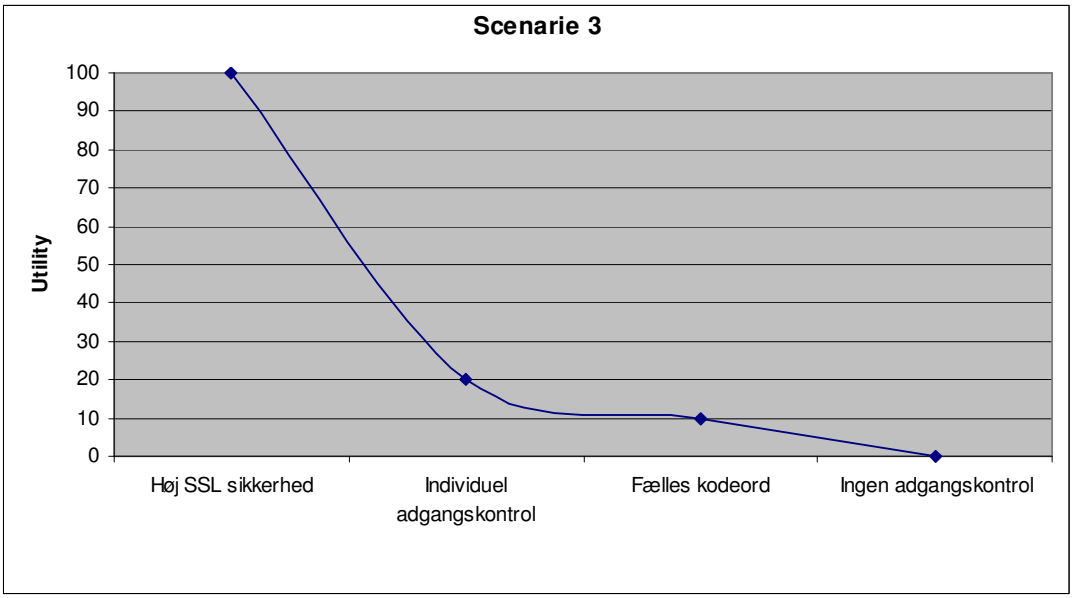


I scenarie 2 er der også 4 muligheder. I HS-07 er der ikke nogen opdeling af administratorer og brugere. Man kan vælge at lave en enkelt administrator med et tilhørende password. Dette er ikke særlig fleksibelt

og kræver at alle der skal kunne udføre administrator-funktioner kender passwordet. Ofte vil det dog være tilstrækkeligt at en enkelt bruger af systemet kan udføre administrator-funktionerne. Et lidt bedre alternativ er at alle brugere kan få tildelt en rolle som almindelig bruger eller administrator. Dvs. der kan være et vilkårligt antal brugere der kan have adgang til administrator-funktionerne. Den sidste løsning giver fuld konfigurérbarhed så en hvilken som helst bruger kan få tildelt rettigheder til hver funktion individuelt. Dvs. hver bruger kan have adgang til et vilkårligt antal funktioner. Denne løsning kræver meget opsætning af brugerne, men giver til gengæld mulighed for at man kan vælge fx hvilke brugere der skal have adgang til at slå alarmen fra.



Sikkerheden er ekstremt vigtig, når brugergrænsefladen giver adgang til kontrol af alarmen. Derfor er det kun den højeste SSL sikkerhed der giver høj værdi. Der er ikke fokuseret på sikkerheden hvis en eventuel trussel får fysisk adgang til gatewayen, da selve gatewayen ikke indeholder fortrolige oplysninger. Hvis man vælger adgang med et fælles kodeord giver det ikke mening at lave flere grupper/brugere i scenarie 2.



Det giver følgende værdier i vores responsmål:

Response goals – Utility Scores					
Scenario	**Votes**	**Worst**	**Current**	**Desired**	**Best**
1	175	0	0	80	100
2	50	0	0	80	100
3	175	0	0	100	100

4.3.1.5 Udvikl strategier/Bestem værdi

Ud fra strategierne kan den forventede værdi aflæses direkte på graferne i afsnit 4.3.1.4:

Strategies - Utilities				
Strategy	**Name**	**Scenarios affected**	**Current utility**	**Expected utility**
1	Kommandolinie UI	1	0	30
2	Webbaseret UI	1	0	80
3	Webbaseret UI med hjælpesystem og aggregerede funktioner	1	0	100
4	1 administrator	2	0	50
5	2 grupper	2	0	80
6	Fuld konf.	2	0	100
7	Fælles kodeord	3	0	10
8	Individuel adgangskontrol	3	0	20
9	Høj SSL sikkerhed	3	0	100

4.3.1.6 Beregn afkast

Det beregnede afkast for hver strategi kan ses i følgende tabel:

Calculated benefit					
Strategy	**Scenarios affected**	**Scenario weight**	**Benefit**	**Normalized benefit**	**Total benefit**
1	1	175	30	5250	5250
2	1	175	80	14000	14000
3	1	175	100	17500	17500
4	2	50	50	2500	2500
5	2	50	80	4000	4000
6	2	50	100	5000	5000
7	3	175	10	1750	1750
8	3	175	20	3500	3500
9	3	175	100	17500	17500

4.3.1.7 Vælg strategi

Estimerede priser for implementation af strategier er brugt til at beregne Return-on-Investment:

Strategies – Utilities				
Strategy	**Strategy Cost**	**Total Benefit**	**Strategy ROI**	**Strategy Rank**
1	500	5250	10,5	8
2	800	14000	17,5	3
3	1500	17500	11,7	7
4	100	2500	25	2
5	250	4000	16	5
6	700	5000	7,1	9
7	100	1750	17,5	4
8	250	3500	14	6
9	400	17500	43,75	1

4.3.1.8 Bekræft resultater

Den prioriterede rækkefølge er sådan:

1. Strategi 9 - Høj SSL sikkerhed
2. Strategi 4 - 1 administrator
3. Strategi 2 - Webbaseret UI
4. Strategi 7 - Fælles kodeord
5. Strategi 5 - 2 grupper
6. Strategi 8 - Individuel adgangskontrol
7. Strategi 3 - Webbaseret UI med hjælpesystem og aggregerede funktioner
8. Strategi 1 - Kommandolinie UI
9. Strategi 6 - Fuld konf.

De 3 højest prioriterede strategier er fint i overenstemmelse med de ønskede niveauer af værdi. Strategi 4 ligger lidt under det ønskede niveau, men det var vurderet at det ville være tilstrækkeligt til et system af denne type, hvor en enkelt administrator ofte er tilstrækkeligt. Analysen viser også at den fulde konfigurérbarhed af brugere er for dyr i forhold til den værdi den giver.

4.3.2 aSQA

Da aSQA metoden i høj grad omhandler samme problemstillinger samt resultat som i H4, er denne del udeladt. Det er dog oplagt at udføre aSQA løbende ifbm. udviklingen af systemet, for at drage nytte af den kontinuere letvægtsmetode, som giver løbende feedback på systemets tilstand, og ikke kun når der skal træffes beslutninger som explicit har implikationer på arkitekturen.

4.4 Overall process driver

Der er brugt en backlog til at holde styr på den overordnede proces. For at give nem adgang til informationerne for hele teamet er denne placeret på en wiki[3].

[3] http://wiki.daimi.au.dk/saip2008/bravo.wiki?cmd=get&anchor=Bravo

5 H5d - Summary and Discussion

"Summarize and discuss your experiences with the design method and in particular with design decisions."

Gruppen har til denne opgave prøvet lave en designmetode til en tænkt organisation. Metoden er blevet prøvet af ved at udføre dele af metoden i praksis. Formålet med opgaven har ikke været at lave en fuld implementation af de ønskede ændringer, men i stedet at afprøve designmetoden til et vist niveau.

Med dette i mente har gruppen undladt at lave kode til besvarelse af opgaven og i stedet fokuseret på at udføre nogle af de foreslåede teknikker.

Det er gruppens vurdering at metoden passer godt til organisationen og at den tid som investeres i formelle metoder er givet godt ud, især da modellen fokuserer på metoder som giver et fornuftigt resultat for en beskeden indsats.

Jf. ovenstående er der ikke lavet udviklingsarbejde på prototyper. Det er dog gruppens vurdering at de er en vigtig del af metoden, da ATAM eller lign. ikke er det. Når ATAM ikke benyttes er der behov for andre metoder til at vurdere om arkitekturen opfylder de stillede kvalitetskrav. Her er det gruppens vurdering at arkitekturprototyper er et stærkt værktøj, da de giver en stor grad af mulighed for direkte at måle om kvalitetskrav er opfyldt.

At arbejde ud fra en beskrevet designmetode betyder at vigtige trin i processen ikke springes over, hvilket igen burde sikre produkter af en højere og især mere ensartet kvalitet.

De artefakter som metoden genererer har i gruppens overbevisning været til god hjælp når designbeslutninger har skullet tages. Arkitekten er således ikke så meget på bar bund, men kan benytte disse til at foretage gode valg og samtidigt få valideret at disse valg faktisk var gode. Desuden understøtter flere af artefakterne udmærket kommunikationen mellem arkitekten og de andre aktører.

Ud over de faktuelle detaljer i at den anvendte Wiki ikke er stærk nok til at håndtere flere personers samtidige arbejde, har det været en god oplevelse at benytte en backlog på denne måde. Der har hele tiden været overblik over de forskellige udeståender, og da Wikien er et letvægtsprodukt ligesom den beskrevne designmetode, har disse fungeret godt sammen.

Dokumentationen af design decisions på en struktureret måde, hvor ikke kun valgene skreves ned, men også alternativer, er en stor hjælp for arkitekten, specielt når projekterne varer i lang tid og det er svært at huske præcis hvorfor en given løsning blev valgt fremfor andre mulige løsninger. Desuden kan produktejeren og udviklerne have gavn af at læse design beslutningerne, og på den måde få indsigt i de beslutninger der er taget.

Designmetoden sikrer at intet glemmes på det overordnede niveau, og backloggen sikrer at det samme gælder for detaljen.

/Bravo

6 References

[1] C. Hofmeister, P. Kruchten, R. Nord, H. Obbink, A. Ran, and P. America.
A general model of software architecture design derived from five industrial approaches. The Journal of Systems & Software, 80(1):106–126, 2007.

[2] P. Kruchten, P. Lago, and H. van Vliet.
Building up and reasoning about architectural knowledge. QoSA, 2006, 2006.

[3] J. Tyree, A. Akerman, and C. Financial.
Architecture Decisions: Demystifying Architecture. Software, IEEE, 22(2):19–27, 2005.

[4] Christensen, H., Corry, A., and Hansen, K. (2004).
An approach to software architecture description using UML. Technical report, Computer Science Department, University of Aarhus.

[5] Bass, L., Clements, P., and Kazman, R. (2003).
Software Architecture in Practice, 2nd Edition, Addison Wesley, 2003.