

Sudden Impact

Processororienterad programmering
(1DTo49) våren 2012. Peer review på
grupp 1 genomförd av grupp 2

Av:

Emil Vikström - 880728-0493

Patrik Broman - 810416-7138

Erik Samuelsson - 890112-0413

A) Helhetsintryck, sammanhang och innehåll

Rapporten ger ett trevligt intryck vid en första genomläsning. De lösningar som valts förklaras ofta lagom övergripande.

Fokus i rapporten ligger tydligt på samtidighet samt på gränsytan mellan Java och Erlang. Detta är lyckat eftersom det är vad projektet främst syftar till. Nackdelen är att avsnittet om "Installation och fortsatt utveckling" verkar fått mindre fokus och saknar en del information såsom vilken Java-implementering som används (Oracle? OpenJDK? Vilken version?) samt information om hur man bygger jar-arkivet. Detta är såklart inte lika intressant vare sig att läsa eller skriva men det är ändå viktigt för att som utomstående kunna göra förändringar i spelet.

Vi saknar en diskussion av de svårigheter gruppen stött på. Uppstod några samtidighetsproblem? Om så var fallet, hur löstes dessa? Kan gruppen tänka sig att det finns buggar som beror på samtidighet och/eller gränsytan Java<->Erlang?

Bild 3 om systemarkitekturen ger värdefull information som kompletterar texten. Denna bild gör det tydligt var uppdelningen sker mellan de två programmeringsspråken. Däremot finns ingen figur som förklarar hur de olika modulerna och objekten hör ihop, vilket gör det aningen svårare att visualisera systemarkitekturen mer detaljerat.

Själva texten i avsnittet "Systemarkitektur" är tyvärr en besvikelse. Avsnittet är alldeles för kort, trots att det i vår mening är bland det mest intressanta. En övergripande förklaring av moduler och beroenden vore önskvärt. Kanske även en diskussion om fördelar och nackdelar med den valda arkitekturen.

I avsnittet "Algoritmer och datastrukturer" är det svårt att förstå vad funktionen "start" gör. Är detta en särskilt intressant del eftersom gruppen valt att kommentera den? Det finns även en del meningslösa stycken, som till exempel "close() - Dödar alla processer i Erlang". Vi hade gärna sett mer övergripande förklaringar av de algoritmer för samtidighet som används. Vore det inte till exempel intressant att förklara hur en duell går till?

Avsnittet "Reflektion" är spännande då det är intressant att läsa vad gruppen lärt sig, men samtidigt är avsnittet delvis svårförståeligt. Det påstås att programmet skulle blivit snyggare och gått snabbare att skriva i ett enda språk. Är det helt sant? Det är klart att man med meddelandeköerna som kommer med J2EE kan bygga message passing-system, men är detta något som varit lättare att lära sig? Om man istället frångått actor-modellen hade det kanske uppstått nya samtidighetsproblem som kunnat ta lång tid att reda ut. I första stycket i inledningen påstås det ju rent av vara en fördel att använda två språk då man kan utnyttja styrkorna i båda. En djupare diskussion av ämnet önskas!

Gruppen antyder att det varit svårt att förstå hur man kommer igång med versionshantering. Vi ifrågasätter detta då de flesta läst kursen "Imperativ och objektorienterad programmeringsmetodik" där versionshantering användes. Det finns även mängder av hjälp att

få via det världsomspännande datornätet Internet, och i nödfall kan man säkert be kursledaren rakt ut om hjälp. Vi tycker att man snarare ska se det som en lärdom att infrastruktur är en ofta oförutsedd tröskel för programmeringsprojekt, och att det kan vara värt att fundera på sådant innan man sätter igång att koda. Det kan vara smart att sätta en projektmedlem på att få upp infrastrukturen och dokumentera hur den används för de andra medlemmarna.

Avslutningsvis vill vi gärna se en förklaring till påståendet, i avsnittet "Programmeringsspråk", att Java lämpar sig dåligt för nätverkskommunikation ur effektivitetssynpunkt. Varför då? Är det en inbyggd svaghet i Java eller beror det på systemarkitekturen? Det är väl bara UI:t som är skrivet i Java?

B) Struktur och disposition

Inledningen ger en övergripande bild av projektet. Vi vill läsa mer, och går in på "Sudden Impact" som på ett naturligt sätt tar över och börjar förklara själva spelet. Därefter tar de tekniska bitarna vid och förklarar på väl avgränsade sätt olika aspekter av systemet. Författarna introducerar först hur programmet är tänkt att fungera, och i senare avsnitt går de in på hur detta uppnåtts. Detta gör det lätt att hänga med i tankegångarna.

Styckeindelning görs på bra ställen. Varje stycke berättar om ett mycket avgränsat område av ämnet. Ibland önskar vi dock att styckena organiserades på ett annat vis, eller i alla fall att det fanns en andra nivå av rubriker. Till exempel hade avsnittet "Programmeringsspråk" tjänat på att Java och Erlang antingen diskuterades var för sig i varsitt rubricerat avsnitt eller att fördelar/nackdelar diskuterades var för sig. I styckeindelningen är det nästan så redan, men det hade gärna fått göras tydligare med rubriker.

I avsnittet "Algoritmer och datastrukturer" dyker det upp en drös kodexempel. Dessa är lite för korta för att egentligen tillföra något till rapporten. Här hade ett eller flera diagram förklarat bättre vad som sker.

"Förslag på förbättringar" är ett ostrukturerat avsnitt. Innehållet är läsvärt men det saknar övergripande struktur. Avsnittet börjar med en punktlista följt av ett antal stycken text. Textstyckena verkar vara ytterligare förslag på förbättringar som inte har med punktlistan att göra (förutom ett par av styckena; förvirrande!). Vissa av dessa stycken har fått någon form av oformaterad rubrik, andra inte. Det blir otydligt var ett avsnitt slutar och det är svårt att få en överblick.

Rapporten följer inte helt den givna mallen då det saknas rubrikenumerering.

C) Språk och formattering

Språket är överlag lättläst. Författarna slår an en berättande ton som inbjuder till läsning. Det är trevligt att läsa rapporten. Mindre trevligt är att det tyvärr finns en hel del stavfel och grammatiska fel.

Ibland har rapporten problem med tempus. Det känns underligt att läsa i futurum när projektet redan genomförts. Exempel (vår fetstil):

- “Utöver detta **förväntas** det bli en utmaning att”
- “det **kommer** ta mycket tid att ta reda på”

Olika tempus används ibland inom samma stycke, till exempel:

- “Målsättningen med projektet **har varit** att få en fungerande prototyp av ett RPG-spel att fungera. Första målet **är** att”
- “Fördelar med den valda implementationen **är** att kommunikationen mellan processer sker väldigt snabbt ... Om ett fel uppstår i Java **kommer detta att** påverka vad som händer i Erlang och vice versa.”

Det andra exemplet är kanske ett gränsfall, men om vi utgår från att fel inträffar så kan det istället stå “Om ett fel uppstår i Java **påverkar** det ...”. Det tycker i alla fall vår grupp vore mer lättläst.

Det finns många fall av underlig meningsbyggnad. Två tips är att det ofta är bra att stryka onödiga ord samt att inte säga flera olika saker i samma mening. Vi ger några exempel nedan, men det finns många fler. Korrekturläs noga!

- “Hur detta kan göras, vad för problem som kan uppkomma och möjligheterna som finns med en sådan sammanslagning.”
 - Detta är inte en fullständig sats. Den ska antingen hänga ihop med den tidigare meningen eller få någon form av omarbetning. Just nu känns det som talspråk.
- “Utvecklat av Ericsson tillåter detta bibliotek att kommunikation sker direkt mellan Erlang och Java”
 - Är det på grund av Ericsson-ursprunget som biblioteket tillåter sådan kommunikation?
- “Skulle implementation gjorts i t.ex. enbart Java skulle trådar komma att användas istället för processer och message passing, men eftersom endast en tråd används i Java finns ingen risk för deadlock, vilket är väldigt positivt!”
 - Det här är en väldigt lång och konstig mening. Den säger alldeles för många saker samtidigt. Till exempel diskuteras både nackdelar med en eventuellt annan implementation *och* en fördel med den nuvarande. Lägg också märke till den obestämda formen av ordet *implementation*.

Ett vanligt stavfel är att ordet "i" skrivs med versal. Det kanske är ordbehandlaren som ändrar automatiskt om språket är inställt på engelska.

Det finns även rena slarvfel och särskrivningar som kan upptäckas vid en korrekturläsning:

- "distrubuted Erlang"
- "skall se **mycket mer smidigare** ut"
- "Eftersom Erlang är ett funktionellt språk lämpade **de** sig väldigt bra"
- "AI upplevelse"
- "Clint Eastwood filmer"
- "rapporten för **denna** projekt"
- "Erlang koden"

Bild 3 saknar numrering och bildtext.