

Projektspecifikation

Daniel Henell, 871015-7515, dhenell@kth.se
Henrik Mattsson 800607-6916, hmatt@kth.se
Handledare: Mads Dam

10 februari 2010

Sammanfattning

Projektets syfte är att utforma ett domänspecifikt språk för att lösa konfigurationsproblem i samband med driftsättning av system i multipla miljöer samt implementera en tolk för detta språk.

1 Bakgrund

Ett och samma system kan behöva köras i ett flertal olika miljöer, t.ex. utvecklingsmiljö, testmiljö, drifttestmiljö och i skarp drift. Mindre skillnader i konfiguration leder då till dubbelarbete när inställningar behöver förändras globalt över samtliga miljöer. Ofta förekommande är att ändringar i konfigurationsfiler inte blir propagerade till varje miljöns konfigurationsfil och på så sätt orsakar problem vid driftsättning eller testning.

2 Existerande lösningar

Macrospråk som t.ex. M4 eller CPP medför ett preprocessorsteg mellan modifikation och driftsättning. Injektion av lokala värden som i fallet Apache Ant[1] är också ett existerande alternativ. Mer eller mindre hårdkodade variabelkontroller som t.ex. Caucho Resin[2], där en och samma konfigurationsfil kan användas både i GPL- och betalversionerna av applikationsservern existerar också. I Visual Studio 2010 används XDT (XML Document Transform) för att manipulera en allmän XML-konfigurationsfil för att anpassa den till olika miljöer. Varje miljö har då en XDT-fil med just de förändringarna som behövs för att gå från den allmänna konfigurationen till att passa för den specifika miljön. Parametrar som inte ändras behåller då sitt standardvärde och på så sätt behövs parametrar inte dupliceras i flera fristående konfigurationsfiler.

3 Problemformulering

Hur kan problemet med konfigurationsfiler för olika miljöer lösas med hjälp av ett domänspecifikt språk, så att redundans i konfigurationsfiler undviks och att ändringar i huvudkonfigurationen propageras till

alla målmiljöer vid driftsättning. Vilka olika implementationstekniker existerar för domänspecifika språk och vilken är mest lämplig för denna tillämpning. Parsergeneratorer såsom Yacc[3] bör därför jämföras med exempelvis Parsec[4]. Övriga frågor att besvara är vilka fördelar ett domänspecifikt språk ger jämfört med generella datamodelleringspråk som såsom XML, JSON och Yaml.

4 Projektplan

Projektet kan delas upp i ett antal disjunkta faser vilka alla kombineras med dokumentation och rapportproduktion.

- * Kravformulering - funktionella krav på språket
- * Bakgrundsforskning
- * Utforma grammatik
- * Utvärdera implementationstekniker, val av densamma
- * Implementation
- * Utvärdering och färdigställande av rapport

5 Tidsplan

- Projektspecifikation 29 jan
- Språkdesign 3 feb
Målet är att producera en grammatik i BNF för språket, en nödvändighet för implementationsfasen.
- Implementation 26 feb
Målsättningen är en fungerande implementation som kan tolka en konfigurationsfil och leverera en samling konfigurationsvärden i lämplig intern datastruktur.
- Mittvägsmöte 10 mar
- Inlämning av uppsats 3 maj
- Exjobbskonferens 10 maj

Referenser

- [1] *Apache Ant Manual*. <http://ant.apache.org/manual/CoreTasks/property.html>.
- [2] *Caucho Resin Manual*. <http://www.caucho.com/resin-3.1/doc/resin-tags.xtp>.
- [3] Stephen Johnson Bell and Stephen C. Johnson. Yacc: Yet another compiler-compiler. Technical report, 1979.
- [4] Daan Leijen University, Daan Leijen, and Erik Meijer. Parsec: Direct style monadic parser combinators for the real world. Technical report, 2001.