

Planning

**Useful feedback in the
Ampersand parser**

Maarten Baertsoen and Daniel S. C. Schiavini

*Open Universiteit Nederland, faculteit Informatica
T61327 - Afstudeerproject bachelor informatica*

October 2, 2014

Contents

1. Introduction	3
1.1. Context	3
1.2. Related work	3
1.3. Document overview	3
2. Objectives	4
3. Domain analysis	5
3.1. Baertsoen	5
3.2. Schiavini	5
4. Results	5
5. Assessment	5
6. Conclusion and next steps	5
A. Appendices	6
A.1. Planning	6
A.1.1. Original planning	6
A.1.2. Planning evaluation	6
A.2. Evaluation	6
A.3. Usage information	6

1. Introduction

1.1. Context

Opdrachtgever Bastiaan Heeren / Stef Joosten - OU Informatica

Research context Modelleren van bedrijfsregels

Taal Haskell

- Refactoren van de source code van een parser uit een open source project
- Verbeteren van de parser-foutmeldingen

1.2. Related work

Onderzoekscontext/related work, plaats van project daarin, mogelijke consequenties voor het onderzoek.

1.3. Document overview

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2. Objectives

Vraagstelling in detail.

In het kader van het Ampersand onderzoeksprogramma is een compiler gemaakt voor bedrijfsregels. Deze compiler is in Haskell geschreven, en is beschikbaar als open source project. Eén van de vraagstukken is de kwaliteit van de parser-foutmeldingen. Daar is nog nooit goed naar gekeken, met als gevolg dat de foutmeldingen niet altijd informatief zijn voor de gebruiker.

De opdracht is om de bestaande parser in z'n geheel te refactoren, met aandacht voor foutmeldingen en pretty-printing. De uitdaging is om zo informatief mogelijke foutmeldingen te genereren. Als randvoorwaarde geldt dat de Haskell code onderhoudbaar moet zijn en goed moet zijn getest. Wanneer het aan de onderhoudbaarheid mankeert, wordt de nieuwe parser niet in productie genomen.

Een ontwikkelstraat op basis van SVN is aanwezig. Dit project zal in een branch worden uitgevoerd. Bij succesvolle afronding zal het ABI-team deze branch in de productiestroom integreren.

- Analyse naar gebruiksvriendelijke foutmeldingen in compilers
- Vergelijken van diverse Haskell bibliotheken om te parsen en te pretty-printen.
- Inventarisatie van technieken en tools in Haskell ter bevordering van de software-kwaliteit
- Analyse van de huidige ontwikkelstraat in relatie tot software engineering principes zoals continuous delivery/integration en het doen van aanbevelingen om de kwaliteit bewaking te vergroten

De software waaraan wordt gewerkt draagt in belangrijke mate bij aan het onderzoek naar bedrijfsregels dat wordt uitgevoerd door Joosten en zijn onderzoeksgroep. De ABI studenten zullen zich voldoende moeten inwerken in dit onderzoek om een bijdrage te kunnen leveren. De opdracht biedt ook de ruimte om onderzoeksresultaten naar gebruiksvriendelijke foutmeldingen en testmethoden in functionele talen (bijv. QuickCheck) in te passen in het project.

Accenten

analyse van een gebruikerscontext er zal enigszins een beeld moeten worden gevormd van het soort gebruiker van Ampersand en hoe deze gebruiker omgaat met de huidige foutmeldingen

ontwerpen van een toepassing bestaande software moet worden aangepast: daarvoor is het noodzakelijk dat eerst het bestaande ontwerp wordt begrepen en dat dit zorgvuldig wordt uitgebreid met het oog op kwaliteit en onderhoudbaarheid

bouwen en implementeren van een toepassing delen van de software zullen herschreven moeten worden

literatuuronderzoek of bureauonderzoek mogelijke oplossingsrichtingen en documentatie van software bibliotheken zullen onderzocht moeten worden

3. Domain analysis

individueel: details van domein en technieken in relatie met het onderzoeksproject academische verantwoording gemaakte keuzen

3.1. Baertsoen

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3.2. Schiavini

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4. Results

beschrijving van het projectresultaat (het softwaresysteem) overzicht van de oplossing, met enkele globale voorbeelden

5. Assessment

reflectie inhoud, resultaat beperkingen in het resultaat/afbakening (op voorhand en praktisch)

6. Conclusion and next steps

conclusies, aanbevelingen voor vervolg / uitbreiding

A. Appendices

A.1. Planning

A.1.1. Original planning

Medio september 2014 -- medio mei 2015

Fase 2	Taakverdeling, planning	1,5 maand (medio sep – eind okt)	20 uur
Fase 3a	Domeinen & technieken	2 maanden (eind sep – eind nov)	50 uur
Fase 3c	Ontwerp & implementatie	5 maanden (eind nov – eind apr)	230 uur
Fase 3d	Documentatie	5 maanden (eind nov – eind apr)	50 uur
Fase 3b	Onderzoekscontext	2 maanden (eind feb – eind apr)	30 uur
Fase 4	Scriptie en eindpresentatie	1,5 maand (eind mrt – medio mei)	20 uur

A.1.2. Planning evaluation

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

A.2. Evaluation

wat zou je de volgende keer anders/beter doen, als groep, eventueel ook individueel

A.3. Usage information

Overige appendices. Doel: alle extra benodigde informatie voor gebruik (wat nodig is vanuit de scriptie)

- individueel: relevante details van domein- en techniekanalyse
- relevante details van architectuurdiagrammen
- relevante code, listings, uitgewerkte oplossingen
- relevante uitgewerkte voorbeelden