

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/29661> holds various files of this Leiden University dissertation

Author: Helvensteijn, Michiel

Title: Abstract delta modeling : software product lines and beyond

Issue Date: 2014-11-12

Samenvatting

Programmeren is een zeer foutgevoelige activiteit. Naarmate er in een software systeem meer en meer features geïmplementeerd worden, zal de vooruitgang van dat systeem steeds langzamer worden. Programmeurs verliezen snel het overzicht als hun code over het hele project verspreid is, en omringd door de code van anderen. Hierdoor worden sneller fouten gemaakt, en is het onvermijdelijk dat de meeste programmeertijd in onderhoud gaat zitten. Dit leidt tot duurdere software die later op de markt komt.

Om te voorkomen dat een software systeem bezwijkt onder zijn eigen complexiteit, zal de code een duidelijke structuur moeten volgen. Manny Lehman (herinnerd als de Vader van Software-evolutie) gaf het volgende als zijn tweede wet van software-evolutie:

“Naarmate een programma evolueert zal de complexiteit ervan toenemen, tenzij deze actief gehandhaafd of verminderd wordt.”

Idealiter willen we alle code met betrekking tot een bepaalde *feature* (ook wel *concern* genoemd) samenvoegen tot één module — genaamd *feature modularizatie* — en code die tot verschillende features behoort van elkaar scheiden — genaamd *separation of concerns*. Maar vele features kunnen niet makkelijk uitgedrukt worden in bestaande programmeer-abstracties. Zulke features noemen we *cross-cutting concerns*. Hun implementatie moet nu eenmaal tot naar verschillende locaties in het project verspreid worden. Modularisatie en ‘separation of concerns’ zijn dus niet makkelijk tot stand te brengen.

Dit proefschrift gaat over *Abstract Delta Modeling (ADM)*, een formele beschrijving die ons helpt deze eigenschappen in software te behalen.

De software engineering discipline die hier het meest bij te winnen heeft is *Software Product Line Engineering (SPLE)*, een relatief nieuwe ontwikkeling. Ik citeer van der Linden, Schmid en Rommes:

“Software product lines vertegenwoordigen misschien wel de spannendste paradigmaverschuiving in de software-ontwikkeling sinds de komst van ‘high-level’ programmeertalen.”

SPLE houdt zich bezig met de ontwikkeling en het onderhoud van *meerdere* software systemen tegelijk, elk in bezit van een andere (maar vaak overlap-pende) verzameling van features — een vorm van *mass customization*. Aan

de implementatie van een product line stellen we wel extra eisen. Het voldoet niet meer dat de code van een feature gescheiden en modulair is; het moet ook *componeerbaar* zijn, en goed omgaan met de aanwezigheid of afwezigheid van andere features. We moeten uit een verzameling beschikbare features een selectie kunnen maken, en de bijbehorende software mechanisch voor ons kunnen laten genereren. Dit proces staat bekend als *automated product derivation*. Dit is een ander gebied waarin ADM van dienst kan zijn.

Dit proefschrift komt uit het Europese HATS project. Het representeert een formele basis voor *delta modeling*, de techniek die gekozen was door HATS voor het uitdrukken van software variabiliteit. Voor dit doel gebruikt het (onder andere) abstracte algebra, modale logica, operationele semantiek en Mealy machines, en legt het geleidelijk de bruggen tussen deze verschillende disciplines. De hoofdstukken van het proefschrift geven een breed overzicht van het ADM framework, zowel als de mogelijkheden van dit framework en verscheidene praktische applicaties, en legt hiermee een fundering voor verder onderzoek en ontwikkeling.