Onderzoekcontext MDD

ABI team 30

Joop van de Heijning

Open Universiteit

Onderzoekcontext MDD

**Introductie**

In dit verslag worden verbanden tussen de onderzoekcontext van dit project met Model Driven Development (MDD) besproken. Het doel is om meer inzicht te krijgen in de problemen en eventueel gevonden oplossingen die in de literatuur omtrent dit vakgebied te vinden zijn. Dit inzicht kan specifieke nieuwe requirements opleveren voor ons project, of dit inzicht kan zelfs laten zien welke nieuwe wegen ons project in de onderzoekcontext geopend heeft.

Gezien de aard van het project (op bachelorniveau) en de beperkte ruimte in dit verslag is er voor gekozen om de bestudering van de literatuur te beperken tot die onderwerpen die tot de bredere basis van MDD gerekend kunnen worden. Ook bleek na deze beperking het domein nog voldoende vruchtbaar te zijn voor onderzoek. Die onderwerpen die relevant leken zijn hieronder besproken.

In een consult zijn vragen aan de onderzoeker gesteld naar aanleiding van de bevindingen in hoofdstukken Gebruik maken van bestaande systemen, Modellen en standaarden, Annotaties, en Low ceiling. Diens antwoorden zijn aan het einde van deze hoofdstukken in dit verslag verwerkt.

**Automatisering en grafische User Interface**

Eén van de aantrekkelijkste aspecten van MDD is wellicht de automatisering van het schrijven van code. Het verhoogt de productiviteit van producenten van software en verbetert de betrouwbaarheid van het eindproduct (Selic, 2003).

De meeste verkopers van MDD tools hebben zich geconcentreerd op het automatisch produceren van code vanuit modellen (Atkinson & Kühne, 2003). Dit heeft wel tot gevolg dat er een drempel tot gebruik opgeworpen wordt: Programmeurs zijn gedwongen om een nieuwe taal te leren voor het specificeren van de modellen (Myers, Hudson, & Pausch, 2000). Gelukkig kunnen de meeste modellen gebouwd worden in een grafische omgeving (Vanderdonckt, 2008).

Grafisch modelleren sluit goed aan bij het menselijke visuele perceptievermogen en wordt dan ook gezien als één van de fundamenten van de ondersteuning voor MDD. In het bijzonder wordt het gebruik van object georiënteerd modelleren aanbevolen als technologie met de beste staat van dienst (Atkinson & Kühne, 2003).

Fresnel Forms[[1]](#footnote-1), het eindproduct van dit afstudeerproject, bouwt voort op MDD Fresnel, een plug-in voor Protégé [[2]](#footnote-2) (Brenninkmeijer & Zwanenberg, 2014). Deze tool produceert automatisch code (zowel platform-onafhankelijk als –specifiek in de door ons uitgebreide versie) en maakt gebruik van een UML klassediagrammen-achtige Graphical User Interface voor het specificeren van het model. Dit sluit dus goed aan bij bovenstaande bevindingen uit de literatuur.

**Gebruik maken van bestaande systemen**

Een verstandige en praktische manier om nieuwe technieken zoals MDD te introduceren is als extensie van een bestaand project. Dit verlaagt de risico’s en maakt gebruik van al gemaakte investeringen (Selic, 2003). Fresnel Forms maakt handig gebruik van het beheren van een ontologie door Protégé. Wellicht dat deze succesvol gebleken aanpak nieuwe wegen kan openen voor onderzoek naar nieuwe tools die hierop voortbouwen?

Professor Rutledge merkt in het consult op dat deze aanpak onderzoekstechnisch niets nieuws oplevert, wel dat Frensel Forms in combinatie met Protégé een belangrijke aanwinst voor de praktijk is.

**Modellen en standaarden**

Er wordt verwacht dat User Interfaces (UI’s) op steeds meer verschillende platformen zullen moeten draaien in de toekomst (Van den Berg, Meixner, Breiner, Pleuss, Sauer, & Hussmann, 2010). Daarnaast is er de wens om software-artefacten te beschermen tegen veranderingen op platform-niveau, om zo de levensverwachting van het artefact te verhogen. Vandaar dat ontwikkelaars zo veel als mogelijk het automatisch generen van code vanuit platform-onafhankelijke modellen moeten nastreven (Atkinson & Kühne, 2003).

Een probleem van modellen is dat deze vaak sterk verbonden zijn met het model-gebaseerd systeem en niet kunnen worden geëxporteerd (Vanderdonckt, 2008). Daar tools altijd aan verandering onderhevig zijn, beperkt dit de levensverwachting van het software-artefact. Een technische requirement van een tool is dan ook dat modellen worden opgeslagen in formaten die andere tools ook kunnen gebruiken (Atkinson & Kühne, 2003).

Hier tekent de behoefte aan standaardisatie in formaten zich af. Standaardisatie heeft nog meer voordelen behalve het faciliteren van gebruik door verschillende tools, bijvoorbeeld het codificeren van best practices, aanmoedigen van hergebruik en specialisatie (Selic, 2003).

Het automatisch genereren van modellen gebeurt door mapping functions. Deze geven een verhoging van de productiviteit waarbij de verhoging door alleen het gebruiken van modellen in het software proces bij in het niet valt. Mapping functions zorgen er voor dat kennis ingezet bij het ontwikkelen van een software-artefact niet verloren gaat. Modellen gebruikt in een ontwikkelingsproces kunnen zich zo onafhankelijk ontwikkelen, wat bijdraagt aan een langere levensverwachting van de modellen.

Fresnel Forms maakt gebruik van een platform-onafhankelijk model in een gestandaardizeerde vocabulary Fresnel (Pietriga, Bizer, Karger, & Lee, 2006), met een extensie genaamd OWF. Ook wordt verwacht dat het mogelijk zal zijn om een groot deel van de gebruikte mapping function van Fresnel naar Semantic MediaWiki her te gebruiken voor eventueel een mapping naar een ander platform.

Zal het onderzoek van Professor Rutledge in een vorig artikel (Rutledge, 2013) en het artikel op basis van onder andere dit project bijdragen aan een verdere acceptatie van Fresnel als standaard vocabulary voor UI’s van informatiesystemen gebaseerd op het Semantic Web, zoals Fresnel Forms produceert?

Na de introductie in 2006, zegt Professor Rutledge in het consult, werd Fresnel een korte tijd als veelbelovend gezien. Uiteindelijk werd het niet opgepakt, omdat mensen niet het Semantic Web gingen gebruiken zoals zij het World Wide Web gebruiken. Maar in feite wordt er wel veel gebrowsed op het Semantic Web, wikipedia[[3]](#footnote-3) is namelijk een Semantic Web browser. Met Fresnel Forms op Semantic Forms[[4]](#footnote-4) probeert Professor Rutledge te laten zien dat het toevoegen van informatie en browsen op het Semantic Web efficiënter kan. Wellicht wordt Fresnel zo toch wel relevant.

**Onvoorspelbaarheid**

Een punt van zorg is de onvoorspelbaarheid die MDD met zich meebrengt. Het verband tussen specificatie en uiteindelijk resultaat is niet altijd duidelijk. Een goed voorbeeld van onvoorspelbaarheid in Fresnel Forms is het door ons geïmplementeerde heuristische sorteer-algoritme van Falco Paul (Paul, 2014). Één oplossing is om een expliciete documentatie over de werking van de tool aan te bieden, een andere het geven van de mogelijkheid tot voorvertoning van het resultaat van een toepassing van een regel (Vanderdonckt, 2008).

ABI Team 30 zal aanwijzingen aanbieden op de Fresnel Forms website voor het installeren van Maven Site projectdocumentatie[[5]](#footnote-5). Daarnaast zal de tool goed gedocumenteerd moeten worden in de scriptie.

**Prototyping**

Het geven van een mogelijkheid tot voorvertoning haakt in op het gegeven dat experimenteren één van de meest fundamentele manieren van leren is (Selic, 2003). Specifiek met betrekking tot het ontwerpen van UI’s helpen tools met het automatisch produceren van code. Dit stelt de ontwikkelaar in staat om sneller prototype’s te maken en meer iteraties te doorlopen, wat leidt tot UI’s van hogere kwaliteit (Myers, Hudson, & Pausch, 2000). Ook Selic benadrukt het belang van een snelle doorlooptijd van iteraties voor productiviteit.

Voor het produceren van een prototype van een wiki UI met Fresnel Forms is het nu nog nodig om het geproduceerde platform-specifieke model handmatig te importeren op een MediaWiki server met special:import[[6]](#footnote-6). Het is te verwachten dat het versnellen van dit proces met behulp van een API endpoint[[7]](#footnote-7) de productiviteit van eindgebruikers van Fresnel Forms zal verhogen.

**Annotaties**

Niet alle informatie gerelateerd aan UI objecten kunnen bevat worden in een specifieke tool die alle doeleinden dient. Daarom is er behoefte aan enige steun voor design gebaseerd op annotaties (Vanderdonckt, 2008). Fresnel Forms geeft deze mogelijkheid niet. Het is te overwegen om een optie in de GUI toe te voegen die eindgebruikers de mogelijkheid geeft om commentaar toe te voegen aan lenzen of properties. Zouden deze kunnen worden doorgegeven aan het platform-specifieke model (Fresnel met OWF) in de form van een OWF formatting property?

In het consult geeft Professor Rutledge aan dat zulke annotatie properties al bestaan, RDFS heeft bijvoorbeeld label en comment, ook XML heeft comments. Er is waarschijnlijk geen behoefte aan een optie in de GUI, andere MDD tools hebben deze namelijk ook niet. Een ontwikkelaar kan, indien nodig, annotaties handmatig in de code toevoegen.

**Low ceiling**

Één van de moeilijkst op te lossen problemen van het toepassen van MDD is de kwaliteit van de resulterende UI’s (low ceiling probleem). MDD systemen maken het ook moeilijk om een breed scala van oplossingen te verkennen (wide walls) (Vanderdonckt, 2008). Een optimale ontwikkelmethode zou zijn het zowel ondersteunen van systematische MDD, als meer handmatige informele methoden (Van den Berg, Meixner, Breiner, Pleuss, Sauer, & Hussmann, 2010).

De mogelijkheden tot individualisatie van UI’s die Fresnel Forms biedt zijn beperkt tot opties in de GUI en het invullen van CSS. Er bestaan extensies van MediaWiki die het toelaten om JavaScript code te laden, zoals de JavaScript extensie[[8]](#footnote-8). Zou het interessant zijn om te onderzoeken in hoeverre dit kan helpen om het Fresnel Forms platform-specifieke model verder te individualiseren?

Professor Rutledge geeft in het consult aan dat JavaScript eventueel net als CSS gestructureerd toegevoegd zou kunnen worden aan Fresnel. Maar in feite heeft hij zelf al een oplossing voor het probleem gepresenteerd in 2013. In een gegenereerde MediaWiki sjabloon wordt namelijk gecontroleerd op aanwezigheid van een “InformboxTop”. Indien een wiki-ontwikkelaar deze al had gecreëerd, wordt deze bijgevoegd. (Rutledge, 2013).

**Conclusie**

Het is gebleken dat Fresnel Forms gebruikt maakt van de belangrijkste aspecten van MDD, zoals het gebruik maken van automatisering, modellen en prototyping. Ook problemen van MDD zoals beschreven in de literatuur komen terug in Fresnel Forms, zoals onvoorspelbaarheid en low ceiling. Mogelijke oplossingen voor deze problemen zijn beschreven in de desbetreffende hoofdstukken.

Fresnel Forms laat zien hoe een MDD benadering het maken van informatiesystemen voor het Semantic Web vereenvoudigt. Een belangrijke innovatie van Fresnel Forms is dat de resulterende informatiesystemen het browsen en toevoegen van informatie op het Semantic Web makkelijker maken.

**Referenties**

Atkinson, C., & Kühne, T. (2003). Model-Driven Development: A Metamodeling Foundation. *IEEE Software*, 36-41.

Brenninkmeijer, T., & Zwanenberg, T. (2014). *Protégé-OWL MDD Fresnel Plug-in.* Bachelor’s group project thesis, Open Universiteit.

Myers, B., Hudson, S. E., & Pausch, R. (2000). Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction (TOCHI) - Special issue on human-computer interaction in the new millennium, Part 1*, 3-28.

Paul, F. (2014). *Property Ranking Approaches For Semantic Web Browsers - A Review of Ontology Property Ranking Algorithms.* Master's thesis, Open Universiteit.

Pietriga, E., Bizer, C., Karger, D., & Lee, R. (2006). Fresnel: A Browser-Independent Presentation Vocabulary for RDF. *The 5th Semantic Web Conference.* Athens, Georgia.

Rutledge, L. (2013). From ontology to wiki generating cascadable default fresnel style from given ontologies for creating semantic wiki interfaces. *Workshop on Semantic Web Collaborative Spaces (SWCS2013).*

Selic, B. (2003). The Pragmatics of Model-Driven Development. *IEEE Software*, 19-25.

Van den Berg, J., Meixner, G., Breiner, K., Pleuss, A., Sauer, S., & Hussmann, H. (2010). Model-driven development of advanced user interfaces. *CHI '10 Extended Abstracts on Human Factors in Computing Systems* (págs. 4429-4432). Chicago: ACM.

Vanderdonckt, J. (2008). Model-driven engineering of user interfaces: Promises, successes, failures, and challenges. *Proc. of 5th Annual Romanian Conf. on Human-Computer Interaction ROCHI'2008* (págs. 1-10). Bucharest: Matrix ROM.

1. http://is.cs.ou.nl/OWF/index.php5/Fresnel\_Forms [↑](#footnote-ref-1)
2. http://protege.stanford.edu/ [↑](#footnote-ref-2)
3. http://www.wikipedia.org/ [↑](#footnote-ref-3)
4. https://www.mediawiki.org/wiki/Extension:Semantic\_Forms [↑](#footnote-ref-4)
5. https://maven.apache.org/plugins/maven-site-plugin/ [↑](#footnote-ref-5)
6. http://www.mediawiki.org/wiki/Manual:Importing\_XML\_dumps#Using\_Special:Import [↑](#footnote-ref-6)
7. http://www.mediawiki.org/wiki/API:Main\_page [↑](#footnote-ref-7)
8. http://www.mediawiki.org/wiki/Extension:Javascript [↑](#footnote-ref-8)