



B A C H E L O R A R B E I T

in der Fachrichtung
Wirtschaftsinformatik

T H E M A

Konzeption einer DSL zur Beschreibung von Benutzeroberflächen für profil c/s auf der Grundlage des Multichannel-Frameworks der deg

Eingereicht von:	Niels Gundermann (Matrikelnr. 5023) Woldegker Straße 34 17033 Neubrandenburg E-Mail: gundermann.niels.ng@googlemail.com
Erarbeitet im:	7. Semester
Abgabetermin:	13. Februar 2015
Gutachter:	Prof. Dr. Johannes Brauer
Co-Gutachter:	
Betrieblicher Gutachter:	Dipl.-Ing. Stefan Post Woldegker Straße 12 17033 Neubrandenburg Tel.: 0395/5630553 E-Mail: stefan.post@data-experts.de

Inhaltsverzeichnis

Abbildungsverzeichnis	iv
Tabellenverzeichnis	v
Listings	ix
1 Einleitung	1
1.1 Motivation	1
1.2 Vorgehen	2
2 Hinführung zum Thema	5
2.1 Allgemeine Anforderungen an Benutzeroberflächen von pro- fil c/s	5
2.2 Umsetzung der Benutzerschnittstellen für mehreren Plattfor- men in der deg (Ist-Zustand)	7
2.3 Probleme des Multichannel-Frameworks	8
2.4 Kurzvorstellung der Lösung via DSL	9
3 Domänenspezifische Sprachen	11
3.1 Bedeutung	11
3.2 Abgrenzung zu GPL	11
3.3 Vor- und Nachteile von DSL	11
4 Notwendigkeit einer neuen DSL	13
4.1 Beschreibung der Anforderung an die GUI	13
4.2 Kurzanalyse von Defiziten ausgewählter Sprachen zur Beschrei- bung von Benutzerschnittstellen	13

5	Grobkonzept der Sprache und eines Generators	15
5.1	Notwendige Features der neuen DSL	15
5.2	Notwendige Features eines Generators	15
6	Entwerfen einer DSL zur Beschreibung der GUI in profil c/s	17
6.1	Vorstellung elementarer Beispiele von GUIs aus profil c/s . .	17
6.2	Analyse der Metadaten der GUI	17
6.3	Entwicklung der DSL mit Xtext	17
7	Entwicklung des Generators für das Generieren von Klassen für das Multichannel-Framework	19
7.1	Vorstellung der Architektur für die Beschreibung der Benut- zerschnittstelle	19
7.2	Umsetzung des frameworkspezifischen Generators	19
8	Zusammenfassung und Ausblick	21
	Titel anhang a	xi
	Glossar	xii
	Literaturverzeichnis	xiv

Abbildungsverzeichnis

2.1	Web-Client	6
2.2	Standalone-Client	7
2.3	MC-Framework	8
2.4	neuerAnsatz	9

Tabellenverzeichnis

Listings

Kapitel 1

Einleitung

1.1 Motivation

In der heutigen Zeit werden Webseiten und Programme auf vielen unterschiedlichen Geräten von unterschiedlichen Nutzern ausgeführt. So gut wie die interne Umsetzung einer Anwendung auch sein mag, ist die Benutzeroberfläche immer ein wichtiger Faktor, der für den Erfolg einer Anwendung eine große Rolle spielt. [LW] Damit einher geht die Usability¹ einer Anwendung. Denn eine *schlechte Useability führt zu Verwirrung und Miss- bzw. Unverständnis beim Kunden* [Use12]. Dadurch geht letztendlich Umsatz verloren.

Usability hängt sehr stark mit dem Aufbau der Benutzeroberflächen zusammen. Bei der traditionellen Entwicklung² von Benutzeroberflächen muss aber wie Eingangs erwähnt darauf geachtet werden, dass diese auch auf unterschiedlichen Geräten³. Das hat für den Entwickler in der Regel zur Folge, dass dieser mehrere Graphical User Interfaces (GUI)⁴ bereitstellen muss. Somit werden mehrere GUIs mit unterschiedlichen Toolkits oder Frameworks entworfen. Diese Toolkits haben einen starken imperativen Charakter, sind schwer zu erweitern und sie verhalten sich unterschiedlich abhängig von der speziellen Implementierung. [KB11]

Ein anderer Ansatz zur Beschreibung von Benutzeroberflächen ist das Model-

¹Siehe Glossar: Usability

²Siehe Glossar: Traditionelle UI-Entwicklung

³Desktop, Smartphone, Tablet

⁴Siehe Glossar: GUI

Driven Development. Damit sollen UIs anhand der implementierten Funktionen automatisch erzeugt werden können. Allerdings wird die Darstellung dieser generierten UIs von der Darstellung von traditionell implementierter Benutzerschnittstellen übertroffen. [MHP99].

Eine Überlegung, die sich daraus ergibt, ist, ob man diese beiden Ansätze zur Implementierung von UIs (traditionell und Model-Driven) verbinden kann. In dieser Arbeit wird versucht diese Idee an einem ausgewählten Beispiel umzusetzen. Es geht dabei um die Entwicklung einer neuen Sprache mit der ein Entwickler in der Lage ist, die Abstraktion aus dem Model-Driven Ansatz neben der statischen Beschreibung aus dem traditionellen Ansatz zu nutzen, um das UI allgemein zu beschreiben. Diese Beschreibung soll als Grundlage für unterschiedliche UIs, die auf unterschiedlichen Plattformen ausführbar sind, dienen.

1.2 Vorgehen

Die *data experts GmbH (deg)* hat für die Umsetzung von GUIs auf unterschiedlichen Plattformen eine Lösung entwickelt. In Kapitel 2 dieser Arbeit wird zunächst dieser Ist-Zustand in der deg analysiert und die Probleme aufgezeigt.

Im nachfolgenden Kapitel (3) wird auf Domänenspezifische Sprachen (DSL) eingegangen. Dies ist wichtig, da die Umsetzung einer neuen Sprache zur abstrakten Beschreibung eines GUI auf dem Konzept der DSLs beruht.

Für die Entwicklung von GUI gibt es bereits einige DSLs. Aus diesem Grund wird in Kapitel 4 darauf eingegangen, warum die deg nicht mit den bestehenden DSLs zur GUI-Beschreibung arbeiten sollte.

Die Features, die eine DSL für die Beschreibung von GUIs für die deg benötigt, werden im darauf folgenden Kapitel (5) erläutert. Da mit einer DSL alleine noch keine Umsetzung eines Programms ermöglicht, werden in diesem Kapitel auch die notwendigen Features von Generatoren betrachtet, die zur Generierung von Quellcode benötigt werden.

Darauf aufbauend wird zuerst im Kapitel 6 auf die konkrete Umsetzung der DSL eingegangen. Die GUIs der deg sollten damit ausreichend beschrieben werden können. Eine Prüfung dessen wird später statt finden.

Nachfolgen (Kapitel 7) wird ein Generator für diese DSL entwickelt. Somit kann aus der Beschreibung der GUI Quellcode generiert werden, der in der deg eingesetzt und getestet werden kann.

Zum Abschluss (Kapitel 8) werden die Ergebnisse zusammengefasst und ein Ausblick gegeben.

Kapitel 2

Hinführung zum Thema

Da die deg bereits eine Lösung entwickelt hat, mit der es möglich ist, ein einmal entwickeltes GUI auf mehreren Plattformen darzustellen, muss voerst analysiert werden, warum diese Lösung nicht mehr zukunftstauglich ist.

Bei sämtlichen GUIs wird sich auf GUIs des Programms *profil c/s* bezogen. Profil c/s ist InVeKoS-Programm welches von der deg als Client-Server-Anwendung entwickelt wird.

2.1 Allgemeine Anforderungen an Benutzeroberflächen von profil c/s

Die Anforderung, welche zu der Entwicklung der eingangs kurz erwähnten Lösung geführt hat, ist dass der Client von profil c/s sowohl in Web-Browsern (Web-Client) als auch standalone auf einem PC (Standalone-Client) ausgeführt werden soll. Eine weitere Anforderung ist es, dass beide Clients (Web-Client und Standalone-Client) eine ähnlichen Aufbau haben.

In Abbildung 2.1 und Abbildung 2.2 ist das GUI eines Zuwendungsblattes eines Förderantrags zu sehen. Für den Aufbau sind nur die Tabelle und die darunter stehenden Buttons, sowies das Bemerkungsfeld (im Web-Client auf der rechten Seite und im Standalone-Client in der Mitte) von Bedeutung. Insofern wurde die Anforderung bzgl. des gleichen Aufbaus umgesetzt, auch wenn es beim Vergleich der beiden GUIs im ersten Moment nicht so aussieht.

Abbildung 2.1: Web-Client: Zuewndungsblatt [deG07]

Fördergegenstand mit Fördersatz	ff. Ausgaben lt. Amt [EUR]	Finanzierungsart	Berechneter Bew.betrag [EUR]	Tatsächl. Fördersatz [%]	Abzug [EUR]	Zuwendung lt. Amt [EUR]
Erweiterung vereinseigener Sportstätten - 75,0...	50.000,00	A	37.500,00	75,00	0,00	37.500,00
Ausnahmen - 30,00%	20.000,00	A	6.000,00	30,00	0,00	6.000,00
Neubau kommunaler Sportstätten - 75,00%	90.000,00	A	67.500,00	75,00	0,00	67.500,00
Modern. vereinseigener Sportstätten - 75,00%	80.000,00	A	60.000,00	75,00	0,00	60.000,00
Instand. vereinseigener Sportstätten - 75,00%	50.000,00	A	37.500,00	75,00	0,00	37.500,00
Gesamt	290.000,00		208.500,00	71,90	0,00	208.500,00

Abbildung 2.2: Standalone-Client: Zuwendungsblatt [deG07]

Im folgenden Kapitel wird erklärt, wie die deg diese Anforderungen umgesetzt hat.

2.2 Umsetzung der Benutzerschnittstellen für mehreren Plattformen in der deg (Ist-Zustand)

Für die Umsetzung dieser Anforderungen wäre es möglich gewesen für den Web-Client und dem Standalone-Client separate GUIs zu entwickeln. Die deg hat jedoch eine Lösung erarbeitet mit der es möglich ist, ein einmal beschriebenes GUI auf mehrere Plattformen zu portieren. Das reduziert den Aufwand zur Entwicklung neuer GUIs. Die Lösung der deg ist das *Multichannel-Framework*.

Innerhalb dieses Frameworks werden die GUIs mittels so genannter *Präsentationsformen* beschrieben. Die Architektur des Multichannel-Frameworks ist Abbildung 2.3 zu entnehmen. Daraus wird deutlich, dass aus Präsentationsformen mithilfe der *Component-Factories* GUIs erzeugt werden, die

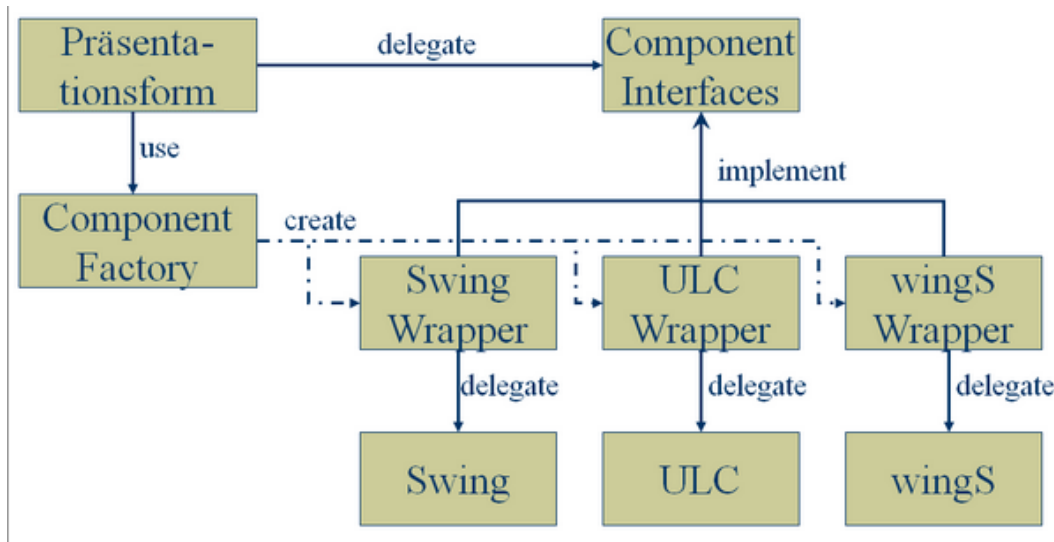


Abbildung 2.3: Architektur des Multichannel-Frameworks [Maa07]

auf unterschiedlichen Frameworks basieren¹. Somit ist die deg in der Lage ihre GUIs für das *Swing*-Framework und für das *wingS*-Framework mit nur einer GUI-Beschreibung zu erzeugen.

2.3 Probleme des Multichannel-Frameworks

Beim Einsatz des Multichannel-Frameworks treten jedoch zwei große Probleme auf. Das erste Problem bezieht sich auf die integrierten Frameworks (Swing und wingS). Beide Frameworks sind veraltet und werden nicht mehr gewartet. Um auch in der Zukunft den Anforderungen der Kunden nachkommen zu können müssten beide Frameworks von der deg weiterentwickelt werden, da diese Aufgabe aus unterschiedlichen Gründen von den Entwicklern der Frameworks nicht mehr wahrgenommen wird. Eine andere Möglichkeit wäre es, wenn die deg andere und modernere Frameworks einsetzt um den nötigen Support der Framework-Entwickler nutzen zu können.

Das Multichannel-Framework ist in der Theorie so konzipiert, dass es leicht sein sollte neue Frameworks zu integrieren (siehe Abbildung 2.3. In der Praxis hat sich jedoch gezeigt, dass es nicht so einfach ist. Das Problem, welches bei der Integration neuer Frameworks aufkommt, ist, dass sich das

¹Hier: Swing, ULC und WingS. Wobei ULC bei der deg nicht mehr im Einsatz ist.

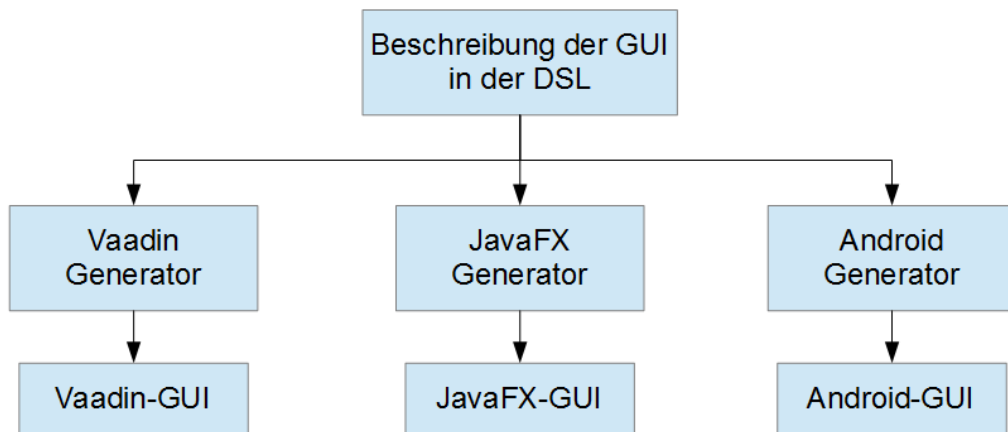


Abbildung 2.4: DSL-Ansatz für gleich GUIs auf unterschiedlichen Plattformen

Multichannel-Framework sehr stark an Swing orientiert und die GUIs vor allem vom GridBagLayout stark beeinflusst sind. Ein solches Layout steht nicht in allen Frameworks zur Verfügung. Da die Beschreibung über ein solches Layout statt findet und der Aufbau der GUIs unterschiedlicher Frameworks per Anforderung gleich sein soll, ist es die Verfügbarkeit eines GridBagLayouts somit eine Voraussetzung für die Intergration in das Multichannel-Framework.

2.4 Kurzvorstellung der Lösung via DSL

Aufgrund der nur schwer machbaren Integration neuer Frameworks in das bestehende Multichannel-Framework und der Tatsache, dass die derzeit genutzten Frameworks (Swing und wingS) veraltet sind, wird ein neuer Ansatz für die Umsetzung von GUIs auf unterschiedlichen Plattformen gesucht.

Der neue Ansatz basiert auf der folgenden Idee. Die GUIs sollen weiterhin nur einmal beschrieben werden sollen. Diese Beschreibung soll über eine DSL erfolgen und sich nicht an bestehende Frameworks orientieren. Grund dafür ist, dass ansonsten die Gefahr besteht, dass langfristig betrachtet mit diesem Ansatz das gleiche Problem auftritt wie beim Multichannel-Framework. Aus der Beschreibung der GUIs wird ein Generator speziellen Quellcode erzeugen, der sich auf entsprechenden Plattformen ausführen

lässt. Für jedes eingesetzte Framework muss somit ein eigener Generator entwickelt werden. Abbildung 2.4 bildet die aus dieser Idee resultierende Architektur ab².

²Hier: Vaadin als Web-Framework, JavaFX als Framework für den Standalone-Client und Android als Repräsentant für einen möglichen Mobile-Client

Kapitel 3

Domänenspezifische Sprachen

3.1 Bedeutung

3.2 Abgrenzung zu GPL

3.3 Vor- und Nachteile von DSL

Kapitel 4

Notwendigkeit einer neuen DSL

- 4.1 Beschreibung der Anforderung an die GUI**
- 4.2 Kurzanalyse von Defiziten ausgewählter Sprachen zur Beschreibung von Benutzerschnittstellen**

Kapitel 5

Grobkonzept der Sprache und eines Generators

5.1 Notwendige Features der neuen DSL

5.2 Notwendige Features eines Generators

Kapitel 6

Entwerfen einer DSL zur Beschreibung der GUI in profil c/s

DSL

- 6.1 Vorstellung elementarer Beispiele von GUIs
aus profil c/s
- 6.2 Analyse der Metadaten der GUI
- 6.3 Entwicklung der DSL mit Xtext

Kapitel 7

Entwicklung des Generators für das Generieren von Klassen für das Multichannel-Framework

- 7.1 Vorstellung der Architektur für die Beschreibung der Benutzerschnittstelle**
- 7.2 Umsetzung des frameworkspezifischen Generators**

Kapitel 8

Zusammenfassung und Ausblick

Titel anhang a

Glossar

DSL ist eine Sprache die für ein bestimmtes Problem optimiert ist.. 2, 17

Förderantrag [...] *ist ein Antrag, den der Begünstigte einreicht, wenn er sich eine Maßnahme fördern lassen möchte* [dat14]. 5

GridBagLayout ist ein Layout Manager innerhalb von Swing, welcher die Komponenten horizontal, vertical und entlang der Grundlinie anordnet. Dabei müssen die Komponenten nicht die gleiche Größe haben [Oraa]. 9

GUI ist die Schnittstelle zwischen dem Benutzer und dem Programm. 1

InVeKoS ist die Abkürzung für Integriertes Verwaltungs- und Kontrollsystem. Mit einem solchen Sysmten wird im allgemeinen sichergestellt, dass die durch den Europäischen Garantiefonds für die Landwirtschaft finanzierten Maßnahmen ordnungsgemäß umgesetzt wurden. Im speziellen bedeutet dies die Absicherung, von Zahlungen, die korrekte behandlung von Unregelmäßigkeiten und das wieder Einziehen von zu unrecht gezahlter Beiträge [Gen14]. 5

Swing ist ein UI-Framework für Java Applikationen [Orab]. xiii, 8, 9

Traditionelle UI-Entwicklung Bei der traditionellen UI-Entwicklung wird mit traditionellen UI-Toolkits gearbeitet. Bei diesen Toolkits wird Aufbau der GUI genau beschrieben. Für die Interaktion mit den UI-Widgets, werden Listener implementiert, die auf andere Events reagieren, die von anderen Widgets erzeugt generiert wurden. Events können zu unterschiedlichen Zeitpunkten generiert werden und es wird nicht

festgelegt in welcher Reihenfolge sie bei anderen Widgest ankommen.
[KB11]. 1

Usability beschreibt die Nutzerfreundlichkeit einer GUI, sowie auch die Nutzerfreundlichkeit einer Software. 1

wingS ist ein Framework für die komponentenorientierte von Webapplikationen [Sch07]. 8, 9

Zuwendungs-Berechner ist ein Werkzeug innerhalb von profil c/s. *Mit diesem Werkzeug kann der Sachbearbeiter die Zuwendung, die dem Antragsteller bewilligt werden soll, nach einem standardisierten Verfahren berechnen (siehe Abschnitt "Algorithmen"). Das Ergebnis wird im Zuwendungsblatt dokumentiert, das auch später mit demselben Werkzeug angesehen werden kann* [deG07]. xiv

Zuwendungsblatt ist die grafische Dokumentation der Ergebnisse des Zuwendungs-Berechners innerhalb von profil c/s. xiv, 5

Literaturverzeichnis

- [dat14] DATA EXPERTS GMBH: *Förderantrag*. Profil Wiki der deg, März 2014. Zuletzt eingesehen am 02.12.2014.
- [deG07] GMBH DATA EXPERTS: *Detailkonzept ELER/i-Antragsmappe*, Januar 2007. Letzte Änderung am 01.12.2014.
- [Gen14] GENERALDIREKTION LANDWIRTSCHAFT UND LÄNDLICHE ENTWICKLUNG: *Das Integrierte Verwaltungs- und Kontrollsystem (InVeKoS)*. URL: http://ec.europa.eu/agriculture/direct-support/iacs/index_de.htm, November 2014. Zuletzt eingesehen am 02.12.2014.
- [KB11] KRISHNASWAMI, NEELAKANTAN R. und NICK BENTON: *A Semantic Model for Graphical User Interfaces*. Microsoft Research, September 2011. Verfügbar unter URL:.
- [LW] LU, XUDONG und JIANCHENG WAN: *Model Driven Development of Complex User Interface*. Doktorarbeit, Shandong University. Verfügbar unter URL:.
- [Maa07] MAASS, DIRK: *JWAMMC - Das Multichannel-Framework der data-experts gmbh*. Vortrag, Dezember 2007.
- [MHP99] MYERS, BRAD, SCOTT E. HUDSON und RANDY PAUSCH: *Past, Present and Future of User Interface Software Tools*. Doktorarbeit, Carnegie Mellon University, 1999.
- [Oraa] ORACLE: *Class GridBagLayout*. URL: <https://docs.oracle.com/javase/7/docs/api/java/awt/GridBagLayout.html>. Zuletzt eingesehen am 02.12.2014.

- [Orab] ORACLE: *Swing*. URL: <https://docs.oracle.com/javase/jp/8/technotes/guides/swing/index.html>. Zuletzt eingesehen am 02.12.2014.
- [Sch07] SCHMID, BENJAMIN: *Get your wingS back!* URL: <http://jaxenter.de/artikel/Get-your-wingS-back>, Dezember 2007. Zuletzt eingesehen am 02.12.2014.
- [Use12] USERLUTIONS GMBH: *3 Gründe, warum gute Usability wichtig ist*. URL: <http://rapidusertests.com/blog/2012/04/3-gute-grunde-fuer-usability-tests/>, April 2012. Zuletzt eingesehen am 01.12.2014.