



## 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

<b>Abbildungsverzeichnis</b>	<b>iv</b>
<b>Tabellenverzeichnis</b>	<b>v</b>
<b>Listings</b>	<b>ix</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Vorgehen . . . . .	2
<b>2 Hinführung zum Thema</b>	<b>3</b>
2.1 Allgemeine Anforderungen an Benutzeroberflächen von pro- fil c/s . . . . .	3
2.2 Umsetzung der Benutzerschnittstellen für mehreren Plattfor- men in der deg (Ist-Zustand) . . . . .	5
2.3 Probleme des Multichannel-Frameworks . . . . .	6
2.4 Kurzvorstellung der Lösung via DSL . . . . .	7
<b>3 Domänenspezifische Sprachen</b>	<b>9</b>
3.1 Bedeutung . . . . .	9
3.2 Abgrenzung zu GPL . . . . .	9
3.3 Vor- und Nachteile von DSL . . . . .	9
<b>4 Notwendigkeit einer neuen DSL</b>	<b>11</b>
4.1 Beschreibung der Anforderung an die GUI . . . . .	11
4.2 Kurzanalyse von Defiziten ausgewählter Sprachen zur Beschrei- bung von Benutzerschnittstellen . . . . .	11

---

<b>5</b>	<b>Grobkonzept der Sprache und eines Generators</b>	<b>13</b>
5.1	Notwendige Features der neuen DSL . . . . .	13
5.2	Notwendige Features eines Generators . . . . .	13
<b>6</b>	<b>Entwerfen einer DSL zur Beschreibung der GUI in profil c/s</b>	<b>15</b>
6.1	Vorstellung elementarer Beispiele von GUIs aus profil c/s . .	15
6.2	Analyse der Metadaten der GUI . . . . .	15
6.3	Entwicklung der DSL mit Xtext . . . . .	15
<b>7</b>	<b>Entwicklung des Generators für das Generieren von Klassen für das Multichannel-Framework</b>	<b>17</b>
7.1	Vorstellung der Architektur für die Beschreibung der Benut- zerschnittstelle . . . . .	17
7.2	Umsetzung des frameworkspezifischen Generators . . . . .	17
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>19</b>
	<b>Titel anhang a</b>	<b>V</b>
	<b>Glossar</b>	<b>VI</b>
	<b>Literaturverzeichnis</b>	<b>VIII</b>

# Abbildungsverzeichnis

2.1	Web-Client . . . . .	4
2.2	Standalone-Client . . . . .	5
2.3	MC-Framework . . . . .	6
2.4	neuerAnsatz . . . . .	7



# 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 Usability einer Anwendung immer ein sehr wichtiger Aspekt. Denn eine *schlechte Useability führt zu Verwirrung und Miss- bzw. Unverständnis beim Kunden* [Use12]. Dadurch geht letztendlich Umsatz verloren.

Bei der Entwicklung von Benutzeroberflächen muss aber wie Eingangs erwähnt darauf geachtet werden, dass diese auch auf unterschiedlichen Geräten<sup>1</sup>. Das hat für den Entwickler in der Regel zur Folge, dass dieser mehrere GUI bereitstellen muss. Somit werden mehrere GUIs entworfen, die dieselben Aspekte abbilden sollen. Von daher stellt sich die Frage, ob es nicht möglich ist, ein GUI in einer speziellen Sprache so abstrakt zu beschreiben, dass einmal beschrieben für alle unterschiedlichen Geräte exportiert werden kann.

In dieser Arbeit wird versucht diese Idee an einem Beispiel umzusetzen. Das vorgesehene Beispiel wird in Kapitel 2 erklärt. Vorerst wird jedoch die Vorgehensweise zur Umsetzung der Idee in dieser Arbeit erläutert. [MSL<sup>+</sup>13]

---

<sup>1</sup>Desktop, Smartphone, Tablet

## 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
<b>Gesamt</b>	<b>290.000,00</b>		<b>208.500,00</b>	<b>71,90</b>	<b>0,00</b>	<b>208.500,00</b>

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

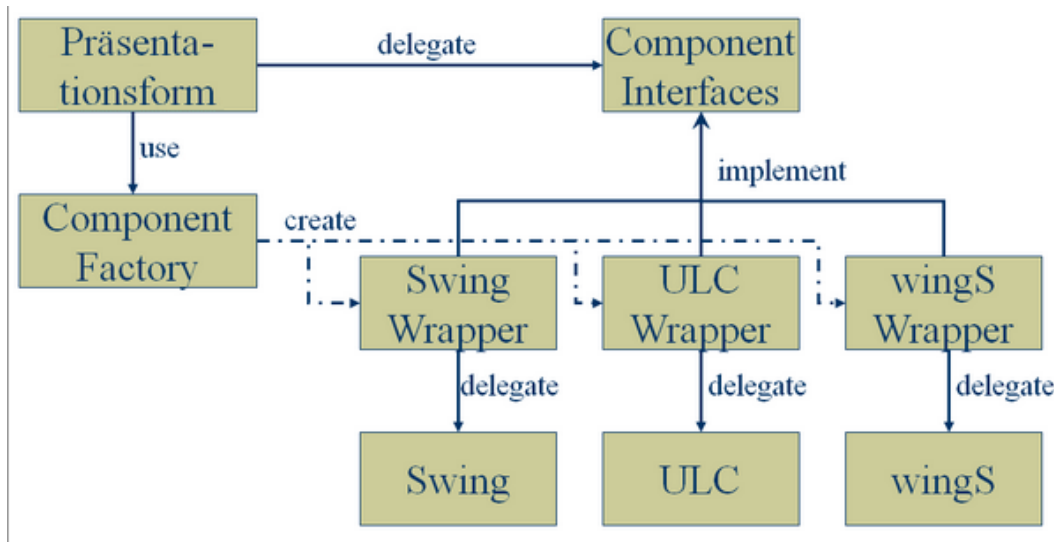


Abbildung 2.3: Architektur des Multichannel-Frameworks [Maa07]

auf unterschiedlichen Frameworks basieren<sup>1</sup>. 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 verwaltet 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

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



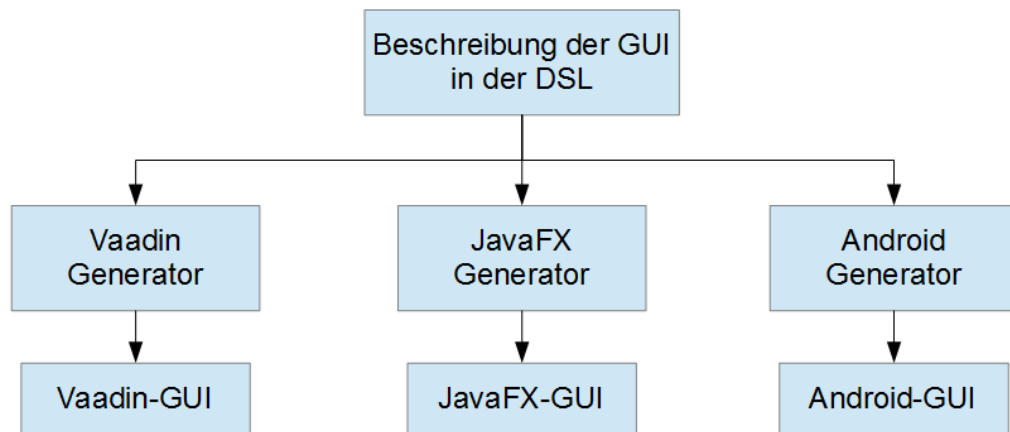


Abbildung 2.4: DSL-Ansatz für gleich GUIs auf unterschiedlichen Plattformen

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 ab<sup>2</sup>.

---

<sup>2</sup>Hier: 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, 15

**Förderantrag** [...] ist ein Antrag, den der Begünstigte einreicht, wenn er sich eine Maßnahme fördern lassen möchte [dat14]. 3

**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]. 7

**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]. 3

**Swing** ist ein UI-Framework für Java Applikationen [Orab]. VII, 6, 7

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

**wingS** ist ein Framework für die komponentenorientierte von Webapplikationen [Sch07]. 6, 7

**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]. VII

**Zuwendungsblatt** ist die grafische Dokumentation der Ergebnisse des Zuwendungs-Berechners innerhalb von profil c/s. 3

# Literaturverzeichnis

- [dat14] DATA EXPERTS GMBH: *Förderantrag*. Profil Wiki der deg, März 2014. Zuletzt eingesehen am 02.12.2014.
- [deG07] GMBH DATA EXPERTS: *Detaillkonzept 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](http://ec.europa.eu/agriculture/direct-support/iacs/index_de.htm), November 2014. Zuletzt eingesehen am 02.12.2014.
- [Maa07] MAASS, DIRK: *JWAMMC - Das Multichannel-Framework der data-experts gmbh*. Vortrag, Dezember 2007.
- [MSL<sup>+</sup>13] MARKUS VOELTER, SEBASTIAN BENZ, LENNART KATS, MATS HELANDER, EELCO VISSER und GUIDO WACHSMUTH: *DSL Engineering*. CreateSpace Independent Publishing Platform, 2013.
- [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.