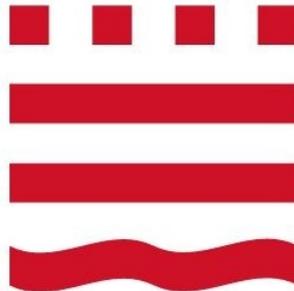


# Modellbasierte Generierung dynamischer Webformulare



## **MASTERARBEIT**

zur Erlangung des Grades eines  
**M. Sc. Wirtschaftsinformatik (FH)**  
des Fachbereichs Wirtschaft der Fachhochschule Brandenburg

1. Prüfer: Prof. Dr. Dietmar Wikarski
2. Prüfer: Dipl. Inf. Christian Fillies

vorgelegt von

<b>Vorname(n)</b>	René	<b>Name</b>	Quoos
<b>geb. am</b>	14.04.1989	<b>in</b>	Potsdam

Brandenburg, den 07. November 2013

## **Abstract**

Traditional forms are used to collect information and to support communication, especially in the public sector. The web offers new possibilities to provide form based services for public administration throughout the internet.

Cit intelliForm is currently a leading tool for development and execution of form-based applications used by the public sector. The approach to create web forms with intelliForm is completely different compared with other tools. Usually web forms are designed with graphical form editors or by use of web technologies like html. IntelliForm allows the semantic modelling of web forms. The model is used to generate java-based web forms which can be deployed and executed on a JBoss application server. Microsoft SharePoint is widely used by companies and by public administration. There is an increasing demand to integrate form-based applications into SharePoint. SharePoint provides the possibility to automate form-based business processes by using workflows. Creating a fully SharePoint integrated form-based application requires the usage of certain Microsoft technologies. InfoPath and InfoPath Forms Services provide an environment to create web-based forms and to publish it on SharePoint. SharePoint Designer offers a simple way to design workflows to process completed forms. The approach of a model-based form generation shows several advantages over the design with traditional form editors like InfoPath. This work is a proof of concept of generating InfoPath forms using a model. Both technologies, intelliForm and InfoPath, are compared to each other to show the advantages of the model-based approach. A secondary aspect is to show how to integrate externally executed forms in SharePoint. The technical feasibility of generating an InfoPath form will be supported by the development of a prototype.

To put it in a nutshell, it is shown that there are several advantages by using models. Those advantages can be leveraged to create form-based applications which are fully integrated within SharePoint.

## Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>III</b>
<b>Abbildungsverzeichnis</b>	<b>VI</b>
<b>1 Einleitung und Motivation</b>	<b>1</b>
<b>2 Grundlagen formularbasierter Anwendungen</b>	<b>5</b>
2.1 Elektronische Formulare und E-Government . . . . .	6
2.2 Webformulare, HTML5 und XForms . . . . .	8
2.2.1 XForms . . . . .	10
2.2.2 HTML5 . . . . .	14
2.3 Nintex Forms und K2 Forms . . . . .	16
2.4 Cit intelliForm . . . . .	18
2.4.1 SemTalk . . . . .	20
2.4.2 Cit intelliForm Composer für Formular-Assistenten . . . . .	21
2.5 Microsoft Info Path und Forms Services . . . . .	24
<b>3 Vergleich des modellbasierten Ansatzes mit InfoPath</b>	<b>28</b>
3.1 Dialogseiten und Ansichten . . . . .	29
3.1.1 Eingabefelder und Steuerelemente . . . . .	30
3.1.2 Beschriftungen . . . . .	32
3.1.3 Unterformulare und wiederholende Abschnitte . . . . .	33
3.1.4 Reihenfolge und Ablaufsteuerung . . . . .	37
3.1.5 Feldvalidierungen, Bedingungen und Regeln . . . . .	39
3.2 Modularisierung . . . . .	47
3.3 Externe Ressourcen und Datenverbindungen . . . . .	49
3.4 Ausführbarer Programmcode . . . . .	50
3.5 Anforderungen im E-Government . . . . .	51
3.6 Vorteile des modellbasierten Ansatzes . . . . .	53

<b>4 Anwendungsszenario mit SharePoint 2013</b>	<b>58</b>
4.1 Beispiel: Brillenshop . . . . .	60
4.1.1 Das Antragsformular . . . . .	61
4.1.2 Publizieren des Formulars auf Microsoft SharePoint 2013 . . . . .	63
4.1.3 Verwendung von SharePoint Workflows . . . . .	65
4.2 Verwendung des SemTalk SiteBuilders . . . . .	68
4.2.1 Ableitung eines Formulars aus einer SharePoint-Struktur . . . . .	69
4.2.2 Integration der cit Formular-/Prozessmodelle in SharePoint . . . . .	71
<b>5 Generierung des Formulars aus dem Modell</b>	<b>75</b>
5.1 Aufbau des InfoPath Formulars . . . . .	75
5.2 Implementierung eines Formulargenerators . . . . .	77
5.2.1 Mapping der cit Modelle auf InfoPath . . . . .	78
5.2.2 Vorstellung des Prototyps . . . . .	79
5.2.3 Einschätzung des Implementierungsaufwandes . . . . .	83
<b>6 Fazit</b>	<b>84</b>
<b>Glossar</b>	<b>VII</b>
<b>Literatur</b>	<b>XII</b>
<b>Anhang</b>	<b>XIII</b>
<b>Ehrenwörtliche Erklärung</b>	<b>XXIII</b>

## Abkürzungsverzeichnis

**BAföG** Bundesausbildungsförderungsgesetz

**CSS** Cascading Style Sheet

**DFG** Deutsche Forschungsgemeinschaft

**HTML** Hypertext Markup Language

**jsp** Java Server Pages

**qeS** qualifizierte elektronische Signatur

**UI** User Interface

**VB** Visual Basic

**W3C** World Wide Web Consortium

**WF** Windows Workflow Foundation

**www** World Wide Web

**WYSIWYG** What You See Is What You Get

**XAML** Extensible Application Markup Language

**XML** Extensible Markup Language

**XSD** XML Schema Definition

**XSL** Extensible Stylesheet Language

**XSLT** XSL Transformation

## Abbildungsverzeichnis

1	Beispiel für HTML-Formular . . . . .	10
2	Beispiel für HTML-Formular (Markup) . . . . .	11
3	XForms Model . . . . .	12
4	XForm Model - Einschränkungen . . . . .	12
5	Anzeige der Kontrollelement im <body>-Element . . . . .	13
6	XForm Submit - Markup . . . . .	14
7	Gesendetes XML-Dokument . . . . .	14
8	HTML 5 Beispiel für URL-Eingabefeld mit Firefox . . . . .	15
9	Nintex Forms . . . . .	17
10	K2 Formular-Designer . . . . .	18
11	Komponenten der intelliForm Produktfamilie . . . . .	19
12	SemTalk Oberfläche . . . . .	21
13	Formularerstellung mit dem cit intelliForm Composer für Formular-Assistenten	22
14	Composer - UI . . . . .	22
15	Composer - Beispiel . . . . .	23
16	Attribute (Satz) . . . . .	23
17	Beispiel des fertigen Formular-Assistenten . . . . .	24
18	InfoPath Designer - Oberfläche . . . . .	25
19	InfoPath Designer - Formular . . . . .	26
20	Forms Services . . . . .	26
21	Composer - Sätze und Eingabefelder . . . . .	29
22	InfoPath - Ansichten . . . . .	30
23	Composer - Eingabefelder . . . . .	30
24	InfoPath - Steuerelemente . . . . .	31
25	InfoPath - Eigenschaften des Textfeldes . . . . .	31
26	Composer - Beschriftungen . . . . .	32
27	InfoPath - Überschriften und Beschriftungen . . . . .	33

28	Composer - Unterformular 1 . . . . .	34
29	Composer - Unterformular 2 . . . . .	34
30	Composer - Modell des Unterformulars . . . . .	34
31	Composer - Multiplizität und tabellarische Kurzfassung . . . . .	35
32	InfoPath - Repeating Section - Design Modus . . . . .	36
33	InfoPath - Repeating Section - Preview Modus . . . . .	36
34	Composer - Reihenfolge und Ablaufsteuerung . . . . .	37
35	InfoPath - Ansichten Wechseln über die Ribbon . . . . .	38
36	Eingabeüberprüfung . . . . .	39
37	Composer - Feldvalidierung - Attribute am Eingabefeld . . . . .	40
38	Composer - Bedingungen an Relationen . . . . .	41
39	Composer - Seitenvalidierung mit Regeln . . . . .	41
40	InfoPath - Steuerelement-Eigenschaften (Ribbon) . . . . .	42
41	InfoPath - Ansichtenwechsel . . . . .	43
42	InfoPath - Feldvalidierung . . . . .	44
43	InfoPath - Ein- und Ausblenden von Eingabefelder I . . . . .	45
44	InfoPath - Ein- und Ausblenden von Eingabefelder II . . . . .	46
45	Composer - Mehrfachverwendung von Sätzen und Eingabefelder . . . . .	47
46	Composer - Mehrfachverwendung von Sätzen und Eingabefelder . . . . .	48
47	InfoPath - Datenverbindungen . . . . .	50
48	InfoPath - Digitale Signatur . . . . .	52
49	cit intelliForm Generator . . . . .	59
50	Fallmanagementsystem - Beispiel EU-DLR Berlin . . . . .	60
51	Seitenstruktur in SharePoint . . . . .	61
52	Startseite des Formulars . . . . .	62
53	Angaben zur Person . . . . .	62
54	Angaben zur Brille . . . . .	63
55	Übersicht der Eingaben . . . . .	63
56	Bestätigungsseite . . . . .	64

57	Publiziertes Formular in SharePoint . . . . .	64
58	Brillenshop - Kundensicht . . . . .	65
59	SharePoint Designer - Textbasierte Ansicht . . . . .	66
60	SharePoint Workflow - Visueller Designer . . . . .	67
61	Brillenshop - Sachbearbeiter Sicht . . . . .	67
62	SemTalk Site Builder . . . . .	69
63	Datenaustausch zwischen intelliForm und SharePoint . . . . .	72
64	Modellbasierte Workflowgenerierung . . . . .	74
65	Zusammensetzung der InfoPath Vorlagendatei . . . . .	75
66	Auszug aus template.xml . . . . .	76
67	InfoPath - Export . . . . .	80
68	Mapping - InfoPath Steuerelemente . . . . .	81

## 1 Einleitung und Motivation

Formulare spielen in der heutigen Zeit des Internets und der elektronischen Datenverarbeitung zunehmend eine immer wichtigere Rolle. Im World Wide Web (www) werden Formulare zur Erfassung und Verarbeitung von Daten verwendet. Beispiele für solche Webformulare sind auf fast jeder Webseite anzutreffen, sei es die Eingabe von Anmelde-daten, das Eingeben von Adress- und Kontaktdaten in Onlineshops oder das Ausfüllen der Eingabefelder von Suchmaschinen.

Elektronische Formulare wurden in den letzten Jahren bei der Modernisierung der öffentlichen Verwaltung immer wichtiger. So bieten immer mehr Verwaltungen diverse Dienste im Internet an. Zu diesen Diensten zählt unter anderem das Bereitstellen von Papierformularen in elektronischer Form. Die Online-Services der öffentlichen Verwaltungen gehen jedoch über die bloße Bereitstellung von Papierformularen in elektronischer Form hinaus. Im Vordergrund steht hierbei nicht nur das computergestützte Ausfüllen und Einreichen der Formulare, sondern vielmehr die weitere Bearbeitung der Formulare im gesamten Abwicklungsprozess.

Cit intelliForm ist eine zurzeit führende Lösung zur Entwicklung und Ausführung von formularbasierten Anwendungen im öffentlichen Bereich. Das Spektrum reicht von Bauanträgen über webbasierte BAföG-Anträge bis hin zur Beantragung von Forschungsprojekten bei der Deutsche Forschungsgemeinschaft (DFG), dem führenden Projektträger in Deutschland. Cit intelliForm ermöglicht die Modellierung von Formularen, aus denen im Web ablauffähige Formulare in einer Java/JBOSS basierten Umgebung generiert und mit einer angeschlossenen Fallmanagement-Komponente ausgeführt werden.

Die cit verzeichnet eine zunehmende Nachfrage, bei der Microsoft SharePoint basier-te Formular- und Workflowlösungen in den Vordergrund treten bzw. in die Java Welt integriert werden müssen. Mit Hilfe modellbasierter Formulare und einer auf Windows Workflow basierten Abwicklung von Geschäftsprozessen könnten weitere Geschäftsfelder durch die cit abgedeckt und neue Chancen auf eine bessere Positionierung auf dem Markt geschaffen werden.

Die Besonderheit von cit intelliForm liegt in der grafischen Modellierung der Formularas-sistenten. Die Formulare werden mit Hilfe von Modellen beschrieben. Allgemein definiert ist ein Modell eine zweckorientierte, abstrakte Abbildung eines Systems ([Kra07, S.71]).

Als Vorlage für die Modellbildung dienen vorhandene Papiervorlagen, dessen irrelevante Merkmale abstrahiert (wie zum Beispiel die Gestaltung des Formulars) und die relevanten (wie zum Beispiel die Steuerelemente) im Modell abgebildet werden ([Kra07, S. 76]). Aus dem vollständigen Formularmodell lassen sich anschließend im Web ablauffähige Formulare generieren.

Eine vergleichbare Lösung in der Microsoft/SharePoint-Welt gibt es mit Microsoft InfoPath, InfoPath Forms-Services und dem Microsoft SharePoint Designer. Anders als bei cit intelliForm werden die Formulare nicht aus abstrakten Modellen generiert sondern vielmehr als ein Ebenbild des Papierformulars auf dem Zeichenblatt dargestellt. Diese Art der Erstellung der Formulare ähnelt dem Prinzip "What You See Is What You Get (WYSIWYG)" (deutsch: „Was du siehst, ist das, was du bekommst“). WYSIWYG ist eine Eigenschaft, bei der ein Dokument während der Bearbeitung am Monitor genauso angezeigt wird, wie bei der späteren Ausgabe.

### **Ziel der Arbeit**

Ziel dieser Arbeit ist eine Machbarkeitsanalyse in der evaluiert wird, inwieweit sich Microsoft SharePoint basierte Formulare aus den cit Formularmodellen generieren lassen. Im Vordergrund steht die Herausarbeitung der Vorteile eines modellbasierten Ansatzes gegenüber der Erstellung der Formulare mit Microsoft-Tools. Ein weiterer wichtiger Aspekt ist die Abbildung der Formularinhalte in SharePoint Listen. Es gilt die Stärken und Chancen des modellbasierten Ansatzes als Entscheidungsgrundlage für eine technische Umsetzung aufzuzeigen.

Um die zentrale Fragestellung der Vorteile eines modellbasierten Ansatzes gegenüber der Erstellung der Formulare mit Microsoft Tools zu klären, wird mit Hilfe eines Anwendungsszenarios ein Beispielformular mit cit intelliForm und Microsoft InfoPath erstellt und gegenübergestellt. Für die Vervollständigung des Anwendungsszenarios wird eine entsprechende Seitenstruktur in SharePoint erstellt, das Formular in SharePoint integriert und ein Beispiel-Workflow eingerichtet.

Eine Machbarkeit der technischen Implementierung soll ebenfalls analysiert werden. Dazu ist es notwendig, Anforderungen und Besonderheiten der InfoPath Formulare aufzuzeigen. InfoPath Formulare bestehen aus einem Konstrukt mehrerer XML-Dateien. Eine

Implementierung erfordert das Erstellen der XML-Dateien aus denen das Formular zusammengesetzt ist. Zentrale Fragestellungen für die technische Umsetzung sind: Was gilt es bei InfoPath Formularen zu beachten und wie sind diese aufgebaut? Wie müssen die Inhalte aus den Formulardateien in InfoPath abgebildet werden? Welche Inhalte lassen sich möglicherweise nicht oder nur in anderer Form mit InfoPath umsetzen? Die Machbarkeit der Implementierung soll durch eine prototypische Entwicklung ausgewählter Hauptfunktionalitäten unterstützt werden.

### **Aufbau der Arbeit**

Die Arbeit unterteilt sich im Wesentlichen in sechs Abschnitte. Nach einer Einführung in die Thematik und der Erläuterung der zentralen Zielstellung dieser Arbeit folgt ein Grundlagenkapitel, in dem grundlegende Informationen zum Verständnis der Arbeit bereitgestellt werden. Dazu gehört eine kurze Einführung in die Thematik Webformulare und dessen Besonderheiten zu herkömmlichen Papierformularen. Weiterhin werden erste Ansätze webbasierter Formulare betrachtet. Anschließend werden die Lösungen mit intelliForm und Microsoft InfoPath kurz vorgestellt.

Im dritten Abschnitt wird die Erstellung der Formulare mit intelliForm und InfoPath gegenübergestellt. Ein Beispiel aus der öffentlichen Verwaltung dient als Vorlage zur Erstellung eines Formulars mit intelliForm und InfoPath. Am Ende des dritten Abschnitts soll eine Aussage über mögliche Vorteile eines modellbasierten Ansatzes gegenüber der Erstellung mit Microsoft-Tools getroffen werden.

Eine Einbindung der InfoPath Formulare in SharePoint wird im darauffolgenden vierten Abschnitt beschrieben. Um ein mögliches Szenario einer SharePoint-gestützten Verarbeitung des Formulars aufzuzeigen, wurde ein kleineres Formular in InfoPath entworfen. Dieser Abschnitt beschreibt die Erstellung einer möglichen Seitenstruktur in SharePoint und die Einrichtung eines SharePoint-Workflows. Weiterhin wird diskutiert, inwieweit sich das Formulardateienmodell zur Erstellung der SharePoint Listen und Bibliotheken verwenden lässt.

Im fünften Abschnitt erfolgt die Betrachtung der technischen Realisierbarkeit zur Generierung von InfoPath Formularen aus den intelliForm Formulardateien. Dieser Abschnitt beschäftigt sich mit der Analyse der InfoPath Formularstruktur aus technischer Sicht. Dazu

werden die einzelnen XML-Dateien, aus denen das Formular zusammengesetzt ist, genauer untersucht und dargestellt wie eine technische Implementation umzusetzen wäre. Besonders wichtig ist dabei das Aufzeigen, wie die Elemente des Formulars in InfoPath abzubilden sind und welche Schwierigkeiten dabei auftreten bzw. welche Inhalte sich nicht oder nur bedingt mit InfoPath umsetzen lassen. Die Entwicklung eines Prototyps soll bei der Einschätzung der Umsetzung unterstützen.

Zum Schluss werden die Ergebnisse der Arbeit zusammengefasst und ein abschließendes Fazit mit einem kurzen Ausblick auf den aktuellen Projektstand und einer Aussage ob und wie es mit diesem Ansatz der Formularerstellung weitergehen könnte.

## 2 Grundlagen formularbasierter Anwendungen

Dieser Abschnitt vermittelt grundlegende Kenntnisse zur Thematik dieser Arbeit. Dazu ist es notwendig zu klären, was Formulare eigentlich sind und welchen Nutzen sie haben. Eine Abgrenzung von herkömmlichen Papierformularen zu elektronischen Formularen und Webformularen wird ebenfalls gemacht. Nachdem die theoretischen Grundlagen geklärt sind, wird näher auf die praktische Umsetzung eingegangen. Es werden verschiedene Technologien und Lösungsansätze näher betrachtet. Anschließend werden die zwei Lösungsansätze cit intelliForm und Microsoft InfoPath vorgestellt.

Ein Formular ist ein strukturiertes Dokument, welches freie Felder für Eintragungen enthält ([ITW13a]). Die wohl bekannteste und älteste Form von Formularen sind Papierformulare. Papierformulare kommen fast überall zum Einsatz, sei es für Geldüberweisungen, Kontoeröffnungen, Bestellungen, Erhebungen oder bis hin zu Anträgen bei öffentlichen Behörden. In der öffentlichen Verwaltung haben Formulare eine ganz besondere Bedeutung. Die vielfältigen Aufgaben im öffentlichen Bereich erfordern immer häufiger einen Schriftwechsel zwischen Behörden und Privatpersonen ([Ver09]). Um die Kommunikation zwischen Bürger und Behörde zu erleichtern, werden sehr häufig standardisierte Formulare (Vordrucke) in Papierform verwendet. Diese sind durch den Bürger auszufüllen und können anschließend eingereicht und weiterverarbeitet werden. Für eine erfolgreiche Kommunikation mit Hilfe eines Formulars ist einiges bei der Gestaltung zu beachten, speziell in der öffentlichen Verwaltung. Sie müssen die geforderten Informationen präzise und für den Bürger verständlich erfragen. Oftmals mögen die Leute ein Formular nicht, schon bevor sie es überhaupt gesehen haben ([CJ09, S. 5]). Das liegt zum einen an der Komplexität der Formulare und zum anderen an der allgemeinen Gestaltung bzw. an der Verständlichkeit der Formulare. Die Bundesstelle für Büroorganisation und Bürotechnik (BBB) des Bundesverwaltungsamtes hat eigens zur Erleichterung der bürgernahen Kommunikation ein Handbuch geschrieben. Ein tieferer Einblick in die Besonderheiten der bürgernahen Kommunikation würde den Rahmen dieser Arbeit allerdings sprengen, daher sei an dieser Stelle auf das Arbeitshandbuch des Bundesverwaltungsamtes verwiesen ([Bun02]).

Zusammenfassend ist zu sagen, dass Papierformulare sehr häufig im täglichen Leben zum Einsatz kommen, diese aber oft durch ihre Gestaltung und Komplexität abschreckend wirken können. Sie dienen der Kommunikation zwischen zwei Parteien und zur Erhebung

von Informationen.

### 2.1 Elektronische Formulare und E-Government

Neben den Papierformularen etablieren sich aber immer mehr die elektronischen Formulare oder auch Webformulare. Mit elektronischen Formularen (auch eForms oder e-Formulare) bezeichnet man im Allgemeinen Formulare, mit dessen Hilfe der Datenerfassungsprozess ohne das Vorhandensein von Papierformularen abgewickelt wird. Im Unterschied zu den herkömmlichen Papierformularen lassen sich elektronisch erfasste Daten leichter in computergestützte Systeme integrieren, sowie kostenintensive Medienbrüche und doppelte Datenhaltung vermeiden. Somit lassen sich die gesammelten Informationen direkt in die Geschäftsprozesse einbeziehen, um einzelne Prozessschritte zu automatisieren und die Genauigkeit der Daten zu validieren. Der Zugang zu elektronischen Formularen wird über Webseiten, Portale, mobile Geräte oder spezielle Software-Anwendungen sichergestellt.

Elektronische Formulare bieten im Gegensatz zu ihren Pendant, den Papierformularen, einige Besonderheiten. Sie lassen sich nicht nur besser in automatisierte Prozesse einbinden sondern bieten auch die Möglichkeit dem Ausfüllenden besser beim Ausfüllprozess durch zusätzliche Hinweise und Hilfetexte zu unterstützen. Allerdings werden an elektronische Formulare wiederum andere bzw. besondere Anforderungen gestellt. Mitunter ist es notwendig, Formulare zu archivieren und revisionssicher zu speichern. Soll die elektronische Form die Schriftform<sup>1</sup> ersetzen, so sind diese Formulare zusätzlich mit einer qualifizierten elektronischen Signatur<sup>2</sup> zu versehen. Um dies zu gewährleisten muss eine entsprechende Infrastruktur eingerichtet und Vorkehrungen zur elektronischen Datenhaltung getroffen werden. Auch die Zugänglichkeit solcher e-Formulare ist erschwert, da diese zusätzliche technische Hilfsmittel voraussetzen.

Mit E-Government wird „...die elektronische Abwicklung von Geschäftsprozessen der öffentlichen Verwaltung und Regierung...“ ([Bun13a]) bezeichnet. E-Government stellt einen wichtigen Beitrag zur Modernisierung der öffentlichen Verwaltung dar, um die Kommunikation zu vereinfachen, Prozesse zu beschleunigen und insgesamt die Effizienz und

---

<sup>1</sup> §126 BGB Schriftform u. §126a BGB elektronische Form

<sup>2</sup> §2 SigG

Qualität zu verbessern. Seit dem Jahr 2000 ist E-Government zu einer zentralen Regierungsaufgabe in Deutschland geworden und wird zunehmend weiter ausgebaut, indem immer mehr Dienstleistungen online bereitgestellt und verbessert werden. Neben den strukturellen Entwicklungen gibt es auch einige nennenswerte Infrastrukturprojekte, wie zum Beispiel den neuen Personalausweis als elektronischen Identifikationsnachweis oder die De-Mail zur einfachen Inanspruchnahme verbindlicher behördlicher Dienstleistungen über das Internet ([Bun13a]).

Mit der Verabschiedung des E-Government-Gesetzes am 7. Juni 2013 wurde ein weiterer wichtiger Schritt in Richtung Modernisierung der öffentlichen Verwaltung eingeleitet. Aus dem E-Government-Gesetz gehen unter anderem folgende wichtige Regelungen hervor ([Bun13b]):

- Verpflichtung der Verwaltung zur Eröffnung eines elektronischen Kanals und zusätzlich der Bundesverwaltung zur Eröffnung eines De-Mail-Zugangs
- Grundsätze der elektronischen Aktenführung und des ersetzenden Scannens
- Erleichterung bei der Erbringung von elektronischen Nachweisen und der elektronischen Bezahlung in Verwaltungsverfahren

Weiterhin werden Regelungen für alternative Verfahren zur qualifizierten elektronischen Signatur getroffen. Das bisherige äquivalent zur Schriftform war bislang nur die qualifizierte elektronische Signatur. Als Alternative zur qualifizierte elektronische Signatur (qeS) sind jetzt auch die De-Mail sowie Webanwendungen der Verwaltung mit einer sicheren elektronischen Identifizierung durch die eID-Funktion des neuen Personalausweises ([Bun13b]) möglich.

Die genannten Verfahren und Änderungen senken die Einstiegshürden für elektronische Formulare seitens der strukturellen Voraussetzungen erheblich. Die Einführung der elektronischen Aktenführung legt dabei auch die Datenerfassung auf elektronischem Wege nahe. Immer mehr Behördliche Dienstleistungen sind online erreichbar. Die Bedeutung der e-Formulare nimmt zu und wird zu einem zentralen Kommunikationsmittel zwischen der öffentlichen Verwaltung und Bürgern bzw. Unternehmen. E-Formulare reduzieren nicht nur den Papieraufwand, sondern verbessern die einzelnen Prozesse durch zunehmende Automatisierungsmöglichkeiten. Davon profitieren nicht nur die Bürger, die ihre

Anträge über das Internet ausfüllen und einreichen können, sondern auch die Behörden, die weniger personellen Aufwand für die Erfassung von Formulardaten investieren müssen. Eine doppelte Datenhaltung kann vermieden werden.

### 2.2 Webformulare, HTML5 und XForms

Webformulare sind elektronische Formulare, die sich im Webbrowser öffnen und ausfüllen lassen. C. Jerret und G. Gaffney definieren ein Webformular wie folgt:

You know a form when you see it.

A *form* is a web page that has boxes you can type into.

You might say it's also a form if you can click radio buttons or choose from drop-downs. But it's the „field for me to type into“ that really makes the difference. [CJ09, S. 5]

Nach dieser Definition ist ein Webformular eine Webseite mit Feldern, in denen Eingaben getätigt werden oder aus Auswahlfeldern eine oder mehrere Optionen ausgewählt werden können. Webformulare sind auf fast jeder Webseite zu finden, z.B. eine Anmeldung, ein Kontaktformular oder die Suchfunktion auf einer Webseite. Allgemein ließe sich sagen, je interaktiver eine Seite, desto stärker basiert das Design der Webseite auf Webformularen ([Dub03, S. 1]). Webformulare spielen in der Gestaltung von Webseiten eine immer wichtigere Rolle, denn durch sie wird eine Webseite interessanter. Der Nutzer ist in der Lage, selber mit der Webseite zu interagieren, anstatt nur durch eine Flut von statischen Informationen zu navigieren.

In seinen Anfängen bot das www neue Möglichkeiten, Informationen und Daten, weltweit verfügbar, bereitzustellen. Diese Informationen konnten jedoch von anderen Nutzern nur gelesen werden. Man erkannte das Potenzial von Formularen im Web zum Versenden von Informationen. So wurden erste Techniken entwickelt, um das Web mit Hilfe von Hypertext Markup Language (HTML), als primäre Sprache zur Erstellung von Webseiten, und Webformularen zu gestalten. Eine entsprechende Spezifikation für HTML-basierte Formulare wurde im Jahr 1993 durch das World Wide Web Consortium (W3C)<sup>3</sup>

---

<sup>3</sup> <http://www.w3.org>

entwickelt ([Dub03, S. 2]). HTML-Formulare kommen allerdings schnell an ihre Grenzen. Eines der größten Probleme von solchen Formularen ist die starke Abhängigkeit von Skriptsprachen<sup>4</sup>. In der Praxis sind HTML-Formulare auf diese Skripte angewiesen, um übliche Aufgaben im Zusammenhang mit Formularen zu erledigen. Dazu gehören unter anderem das Markieren von Steuerelementen als Pflichtfeld, das Durchführen von Berechnungen und das Validieren oder das Anzeigen von Fehlermeldungen. Dadurch werden die Dokumente<sup>5</sup> sehr komplex und erfordern eine Wartung, die sowohl zeitintensiv und teuer ist. Eine weitere Hürde ist die Initialisierung von Formulardaten. Das heißt, eine Webseite „merkt“ sich die eingegebenen Daten des Benutzers, um ein wiederholtes Eingeben zu vermeiden. Um ein leeres Formular mit Daten zu füllen, muss ein neues Dokument entweder Stück für Stück neu erstellt oder ein vorhandenes an verschiedenen Stellen ergänzt werden, was sehr viel Rechenleistung erfordert und zum Flaschenhals werden kann. Eine weitere Einschränkung ist die Repräsentation der Encoding-Formate als „flache“ Daten bzw. Name/Werte Paare. Formulare profitieren mehr von einer reicheren Datenrepräsentation. Im Kontext der öffentlichen Verwaltung wird mit dem Ausfüllen eines Formulars, ein Bearbeitungsprozess angestoßen. So ist zum Beispiel ein Antrag für eine Baugenehmigung durch mehrere Instanzen zu bearbeiten bzw. zu prüfen. Bei einem HTML-Formular bedeutet dies, dass auf jeder Stufe, in der das Formular weitergereicht wird, das Datenformat neu interpretiert werden muss. Dieser Konstruktionsfehler macht es schwierig, reine HTML-Formulare in Workflows zu verwenden ([Dub03, S. 14]).

Aufgrund dieser Limitierung ist selbst ein vermeintlich kleines Formular nur mit viel Aufwand und Pflege zu realisieren. Um die Nutzung von Formularen im Web zu fördern hat das W3C einen neuen Ansatz entwickelt. XForms ist eine Generalüberholung der HTML-Formulare von 1993. Das W3C begann die Entwicklung Anfang 2000 und veröffentlichte Ende 2002 die Spezifikation, welche seit August 2003 als Empfehlung gilt. XForms setzt auf die Stärken von XML beim Modellieren, Erfassen und Serialisieren von Benutzereingaben ([Ram04]).

---

<sup>4</sup> Eine Skriptsprache ist eine Programmiersprache zur Realisierung kleinerer Anwendungen oder Anweisungsfolgen. Verwendung im www finden Skriptsprachen wie z.B. JavaScript beim Realisieren von dynamischen Aspekten oder ganzer Web-Anwendungen und werden sowohl client- wie serverseitig eingesetzt ([ITW13b]).

<sup>5</sup> Als Dokument (auch Web-Dokument) wird oft eine einfache HTML-Seite bezeichnet. Oftmals setzt sich das Web-Dokument aus mehreren Dateien zusammen. [And13]

### 2.2.1 XForms

HTML-Formulare wurden zum Rückgrat im eCommerce. Sie haben ihre Wichtigkeit und Notwendigkeit gezeigt, aber auch verschiedene Dinge, die verbessert werden können. XForms wurden anhand der langjährigen Erfahrungen mit HTML-Formularen entwickelt, um die vorhandenen Schwächen zu verbessern. Der Unterschied zu HTML-Formularen ist neben der Verwendung von XML die Trennung der gesammelten Daten von den Steuerelementen, die die einzelnen Werte sammeln. Diese Trennung ermöglicht eine einfachere Wiederverwendung der Formulare. XForms wurde für die Verwendung in XHTML entworfen, ist aber nicht Teil der Sprache und kann auch in anderen Auszeichnungssprachen verwendet werden. Generell ist XForms bemüht, die Internationalisierung, Wiederverwendung, Barrierefreiheit, Nutzbarkeit und Geräteunabhängigkeit zu verbessern.

Um den Unterschied zwischen herkömmlichen HTML-Formularen und XForms deutlich zu machen und einen Eindruck zu geben, wie ein Formular mit XForms aufgebaut ist, folgt an dieser Stelle ein kleines Beispiel. Abbildung 1 zeigt ein HTML-Formular zum Erfassen von Daten für eine Zahlung. Der Nutzer muss eine Zahlungsmethode wählen und für die Zahlungsmethode Lastschrift sind zusätzliche Angaben für die Kontonummer, Bankleitzahl und Bank notwendig.



Zahlungsart auswählen:  Bar  Lastschrift  
KontoNr.:   
BLZ:   
Bank:

Abbildung 1: Beispiel für HTML-Formular

Im herkömmlichen HTML ist das Markup für die zu erfassenden Daten und dessen Darstellung miteinander vermischt (siehe Abb. 2). Das entsprechende Markup für das Formular befindet sich im `<body>`-Tag des HTML-Dokuments.

Mit XForms wird eine Trennung, welche Daten gesammelt werden und dessen Darstellung möglich. Generell lässt sich die XForms Struktur in drei Komponenten aufteilen. Das XForms Modell stellt alle nicht repräsentativen Aspekte der Webanwendung dar. Das Datenmodell enthält eine XML-Instanz, welche Benutzereingaben, Einschränkungen zur Validierung von Eingaben und Metadaten, wie die Eingaben zum Webserver übergeben werden. Das User Interface (UI) stellt alle abstrakten Steuerelemente und Konstrukte

```
8 <form name="input" action="html_form_action.asp" method="get">
9   <table>
10    <tr>
11      <td>Zahlungsart auswählen:</td>
12      <td>
13        <input type="radio" name="payment" value="bar">Bar
14        <input type="radio" name="payment" value="lastschrift">Lastschrift
15      </td>
16    </tr>
17    <tr>
18      <td>KontoNr.:</td>
19      <td><input type="text" name="kontonr"></td>
20    </tr>
21    <tr>
22      <td>BLZ:</td>
23      <td><input type="text" name="blz"></td>
24    </tr>
25    <tr>
26      <td>Bank:</td>
27      <td><input type="text" name="bank"></td>
28    </tr>
29  </table>
30  <input type="submit" value="Submit">
31 </form>
```

Abbildung 2: Beispiel für HTML-Formular (Markup)

für die Benutzeroberfläche bereit. Dieses Vokabular ist so entworfen, um die zugrunde liegende Absicht der Benutzerinteraktion auf dem Ausgabegerät zu erfassen. Die dritte Komponente ist das „Einreichen“ (engl. submit). Diese Komponente steuert welche Daten wo und wie zum Webserver gesendet werden ([Ram04, S. 11]). Um einen Eindruck über den Aufbau eines XForms Formulars zu erhalten, wird das Beispiel aus Abb. 1 beispielhaft umgesetzt.

### XForms Modell

Das XForms Modell wird innerhalb des <head>-Elements im XHTML Dokument platziert. Hier werden alle zu sammelnden Daten aufgelistet. Anders als in HTML ist man hier nicht an einer flachen Datenstruktur gebunden. Die Daten können gekapselt werden. Abbildung 3 zeigt das Datenmodell unseres Beispiels.

Dies sagt einfach, dass wir vier Informationen sammeln und diese an die URL im *action*-Attribut übergeben werden. Bisher wurden noch keinerlei Angaben über mögliche Einschränkungen getroffen. Einschränkungen können mit Hilfe von XML-Schema getroffen werden. In unserem Beispiel ist die Angabe einer Kontonummer, einer Bankleitzahl und des Bankinstituts nur notwendig, wenn die Zahlungsmethode *Lastschrift* ausgewählt wird. Diese Einschränkungen können wie in Abb. 4 direkt im <model>-Element oder in einer XML-Schema Datei festgelegt werden.

```
1 <?xml version="1.0" ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head>
6 <title>XForm</title>
7
8 <model xmlns="http://www.w3.org/2002/xforms">
9   <instance>
10    <Zahlung>
11      <Zahlungsart/>
12      <kontonr/>
13      <blz/>
14      <bank/>
15    </Zahlung>
16  </instance>
17  <submission action="http://example.com/submit method="post" id="submit">
18 </model>
19 </head>
```

Abbildung 3: XForms Model

```
18 <bind ref="/my:Zahlung/kontonr"
19     relevant="/my:Zahlung/Zahlungsart = 'lastschrift'"
20     required="true()"
21     type="xsd:integer"/>
22
23 <bind ref="/my:Zahlung/blz"
24     relevant="/my:Zahlung/Zahlungsart = 'lastschrift'"
25     required="true()"
26     type="xsd:integer"/>
27
28 <bind ref="/my:Zahlung/bank"
29     relevant="/my:Zahlung/Zahlungsart = 'lastschrift'"
30     required="true()"
31     type="xsd:integer"/>
32 ...
```

Abbildung 4: XForm Model - Einschränkungen

### XForms UI

Die Anzeige der Kontrollelemente wird im `<body>`-Element des XHTML Dokuments realisiert (siehe Abb. 5). Im Gegensatz zum reinen HTML-Formular, wird hier nur die Anzeige der Kontrollelemente gesteuert. Die Kontrollelemente werden mit dem `<input>`-Element eingefügt. Damit die angezeigten Kontrollelemente dem richtigen Formularfeld zugeordnet werden, müssen diese an das entsprechende Feld gebunden werden. Dies erfolgt mit Hilfe des `<ref>`-Attributs für das `<input>`-Element. Zusätzlich zur Bindung wird innerhalb des `<input>`-Elements eine Beschriftung (label) gesetzt. Hinweistexte für den Benutzer lassen sich mit dem `<hint>`-Element realisieren. Hinweise können zum Beispiel angezeigt werden, wenn der Benutzer mit dem Mauszeiger auf die Beschriftung des Feldes zeigt.

```
39 <body>
40   <table>
41     <tr>
42       <select1 ref="zahlungsart">
43         <td><label>Zahlungsart auswählen:</label></td>
44         <td>
45           <item>
46             <label>Bar</label>
47             <value>bar</value>
48           </item>
49           <item>
50             <label>Lastschrift</label>
51             <value>lastschrift</label>
52           </item>
53         </td>
54       </select1>
55     </tr>
56     <tr>
57       <td> <input ref="kontonr">
58         <label>KontoNr.</label>
59         <hint>Nennen Sie uns Ihre Kontonummer.</hint>
60       </input>
61     </td>
62   </tr>
63   <tr>
64     <td> <input ref="blz"><label>BLZ</label></input> </td>
65   </tr>
66   <tr>
67     <td> <input ref="bank"><label>Bank</label></input> </td>
68   </tr>
69 </table>
```

Abbildung 5: Anzeige der Kontrollelement im `<body>`-Element

### XForms Submission

Der „Submission“-Teil regelt, wann und wie Daten an dem Webserver weitergegeben werden. Abbildung 6 zeigt wie die „Absenden“-Schaltfläche in unser XHTML-Dokument eingebunden wird. Das ausgefüllte Formular wird als XML-Dokument serialisiert und an

den Webserver zur weiteren Verarbeitung weitergegeben (Abb. 7).

```
68 <input type="submit" value="Submit" />  
69 <label>Submit</label>  
70 </submit>
```

Abbildung 6: XForm Submit - Markup

```
75 <Zahlung>  
76 <zahlungsart>lastschrift</zahlungsart>  
77 <kontonr>1235467789012345</kotonr>  
78 <blz>16050000</blz>  
79 <bank>MBS Potsdam</bank>  
80 </Zahlung>
```

Abbildung 7: Gesendetes XML-Dokument

### Vor- und Nachteile von XForms

XForms basierte Formulare profitieren sehr stark von XML. Die Formulare erhalten dadurch nicht nur eine viel reichere Datenstruktur, sondern lassen sich durch die Verwendung von XML-Schema mit weniger Script-Aufwand und durch standardisierte Anwendungen validieren. Weiterhin sind die Formulare nicht abhängig von einer Auszeichnungssprache bzw. Skriptsprache und lassen sich einfacher wiederverwenden. Dank der Serialisierung des ausgefüllten Formulars als XML-Dokument können Formulare plattformunabhängig weiterverarbeitet werden. Allerdings hat die Verwendung von XForms auch Nachteile. Für die Erstellung eines Formulars sind weiterführende Kenntnisse in XML, XML-Schema und einer entsprechenden Auszeichnungssprache (z.B. XHTML) notwendig. Eine weitere Hürde besteht in der Anzeige solcher Formulare im Webbrowser. Die heute verbreiteten Webbrowser (z. B. Internet Explorer, Mozilla Firefox, Safari usw.) sind ohne Weiteres nicht in der Lage XForms darzustellen. Für die Darstellung solcher Formulare müssen entweder zusätzliche Erweiterungen für den Webbrowser oder ein XForms fähiger Webbrowser installiert werden. Als Alternative gibt es auch Anwendungen, die XForms in XHTML bzw. Java serverseitig umwandeln und die Anzeige in einem normalen Webbrowser ermöglichen.

#### 2.2.2 HTML5

Seit der Entwicklung der letzten HTML Version (HTML 4.01) aus dem Jahr 1999 hat sich das Web stark verändert. Mit HTML 5 wird vom W3C zusammen mit der WHATWG

(Web Hypertext Application Technology Working Group) seit etwa 2006 ein neuer Standard entwickelt. HTML 5 soll neben neuen Funktionen die Notwendigkeit von externen Plug-ins (wie zum Beispiel Flash) oder Skripten reduzieren. Auch in Sachen Formulare bietet HTML 5 einige Neuerungen, die die Verwendung von Skripten reduzieren sollen. Im Wesentlichen werden mit HTML neue Formular-Elemente und Attribute sowie neue Typen für Eingabefelder hinzugefügt. Die neuen Funktionen für Formulare ermöglichen eine bessere Kontrolle der Steuerelemente und der Validierung von Eingaben.

Zur Erinnerung: in früheren HTML Versionen werden Eingaben durch den Benutzer über ein `<input>`-Element mit dem Typ „Text“ erfasst. HTML 5 fügt eine Reihe neuer Typen für `<input>`-Elemente hinzu. So wird zum Beispiel für ein Feld zur Eingabe einer E-Mail aus `<input type="text" name="E-Mail Adresse"></input>` in HTML 5 `<input type="email" name="E-Mail Adresse"></input>`. Neben „email“ gibt es noch weitere Typen wie zum Beispiel „date“, „number“, „time“, „url“ und „search“. Die Typen steuern zum Teil die Darstellung der Steuerelemente, dienen aber auch zur Validierung von Eingaben. So lässt sich zum Beispiel ein Auswahlfeld für ein Datum über den Typ ohne ein aufwendigeres Skript zur Datumsauswahl erstellen. Beim Absenden des Formulars überprüft der Browser anhand des Typs die Eingabe und gibt entsprechende Fehlermeldungen bei Falscheingaben zurück. Abbildung 8 zeigt ein Beispiel vom W3C für ein URL-Feld<sup>6</sup>. Eine browserseitige Eingabevalidierung würde den Aufwand für zusätzliche Skripte reduzieren. Weitere nützliche Neuerungen sind neue Attribute für das `<input>`-Element. So lassen sich zum Beispiel einzelne Felder von der Validierung ausschließen oder als Pflichtfeld markieren.

The screenshot displays a web browser window with a form titled "Add your homepage:". The form contains a text input field with the value "google.de" and a "Daten absenden" button. A tooltip message "Bitte geben Sie eine URL ein." is visible over the input field. Below the form, a note states: "Note: The type='url' is not supported in Internet Explorer 9 and earlier versions." The browser's address bar shows "W3Schools.com - Try it yourself".

Abbildung 8: HTML 5 Beispiel für URL-Eingabefeld mit Firefox

Letztendlich bietet HTML 5 einen guten Ansatz zur Verbesserung von HTML-basierten Webformularen. Das Problem an HTML 5-Formularen ist, dass HTML 5 noch keine of-

---

<sup>6</sup> [http://www.w3schools.com/html/tryit.asp?filename=tryhtml5\\_input\\_type\\_url](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_type_url)

fizielle Empfehlung des W3C ist und sich noch in Entwicklung befindet. Das bedeutet, dass bis zur finalen Empfehlung sich noch einiges ändern kann. Das Hauptproblem bildet die derzeitige Kompatibilität zu den verschiedenen Browsern. Die Feldvalidierungen beim Absenden des Formulars werden durch den Browser ausgeführt. Viele der neu eingeführten <input>-Typen werden von gängigen Browsern noch nicht unterstützt. Nicht unterstützte Eingabefelder werden vom Browser als Typ „text“ interpretiert und nicht validiert. Das hat zur Folge, dass in manchen Browsern das Feld nicht wie gewünscht dargestellt bzw. validiert wird. Dies erfordert zusätzliche Skripte. Nach dem jetzigen Stand würden die Vorteile der neuen Formular-Funktionen in HTML 5 kaum zum Tragen kommen.

### 2.3 Nintex Forms und K2 Forms

Im SharePoint Umfeld gibt es einige Anbieter die SharePoint basierte Lösungen anbieten. Auch im Bereich Formulare und Workflows sind Vertreter vorhanden. Einer der Anbieter ist die Firma Nintex<sup>7</sup>. Nintex hat sich auf SharePoint Workflowlösungen spezialisiert und bietet eine SharePoint Erweiterung, die es ermöglicht SharePoint Workflows direkt in der SharePoint Weboberfläche zu erstellen. Mit Nintex Forms stellt Nintex einen webbasierten Formular-Designer für SharePoint Listen bereit. Abbildung 9 zeigt den Formular-Designer Nintex Forms. Formularelemente werden einfach per Drag & Drop auf der Zeichenfläche platziert.

Mit Hilfe von Nintex Forms lassen sich Listenformulare in ihrer Funktionalität und Benutzbarkeit anpassen. Weiterhin bietet Nintex Forms eine Unterstützung für mobile Geräte. So lassen sich ganz einfach mobile Versionen der Formulare erstellen. Im Vergleich zu InfoPath gibt es allerdings einige Einschränkungen. Nintex Formulare sind zum Ausfüllen über die SharePoint Weboberfläche gedacht und basieren immer auf dahinterliegenden Listen und sollen somit die Standardlistenformulare ersetzen. Die Formulare sind zudem auf eine Seite beschränkt und lassen keine komplexeren Formulare mit mehreren Seiten zu. Als Datenquellen können lediglich SharePoint Listen verwendet werden, also keine externen Datenquellen oder Webservices.

---

<sup>7</sup> [www.nintex.com](http://www.nintex.com)

## 2 Grundlagen formularbasierter Anwendungen

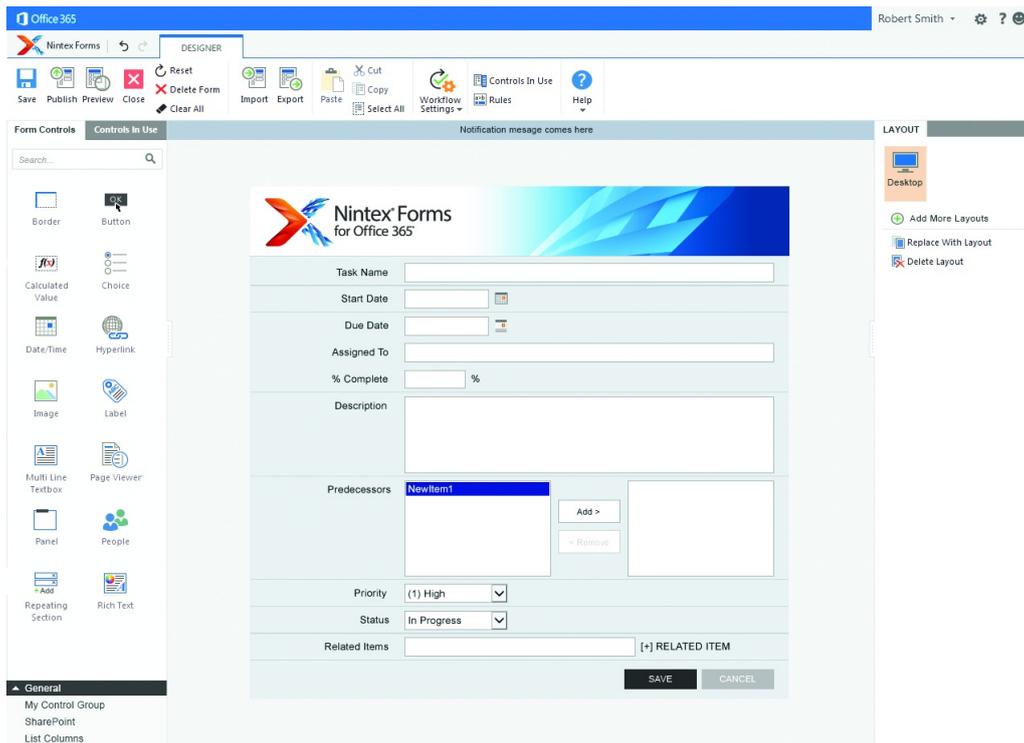


Abbildung 9: Nintex Forms

Ein weiterer Anbieter ist K2<sup>8</sup>. Das Unternehmen stellt eine Plattform zur Entwicklung und Ausführung von Business-Applikationen bereit. Diese lassen sich auch in SharePoint 2013 integrieren. Mit K2 Workflow lassen sich ebenfalls innerhalb der SharePoint Weboberfläche Workflows erstellen und ausführen. Mit K2 Forms deckt das Unternehmen auch die Erstellung von Formularen ab. Diese lassen sich ebenfalls in der Weboberfläche von SharePoint erstellen und verwenden. K2 ermöglicht das Erstellen von einfachen sowie komplexen Formularen. Abbildung 10 zeigt den K2 Formular-Designer.

Im Gegensatz zu Nintex Forms bietet K2 eine eigenständige Formularanwendung die sich nicht nur in SharePoint anzeigen lassen, sondern auch auf anderen Plattform und in HTML5 dargestellt werden können. K2 ermöglicht auch das Anbinden externer Datenquellen aus anderen Systemen. Im Unterschied zu InfoPath basiert K2 auf Webtechnologien wie HTML5, Javascript, CSS und XML und ermöglicht eine einfache Integration in SharePoint.

Nintex Forms ist aufgrund der Beschränkung auf Standardlistenformular nicht als direkter Ersatz zu InfoPath zu sehen. K2 hingegen zeigt sich als Alternative zu InfoPath, da mit K2

<sup>8</sup> [www.k2.com](http://www.k2.com)

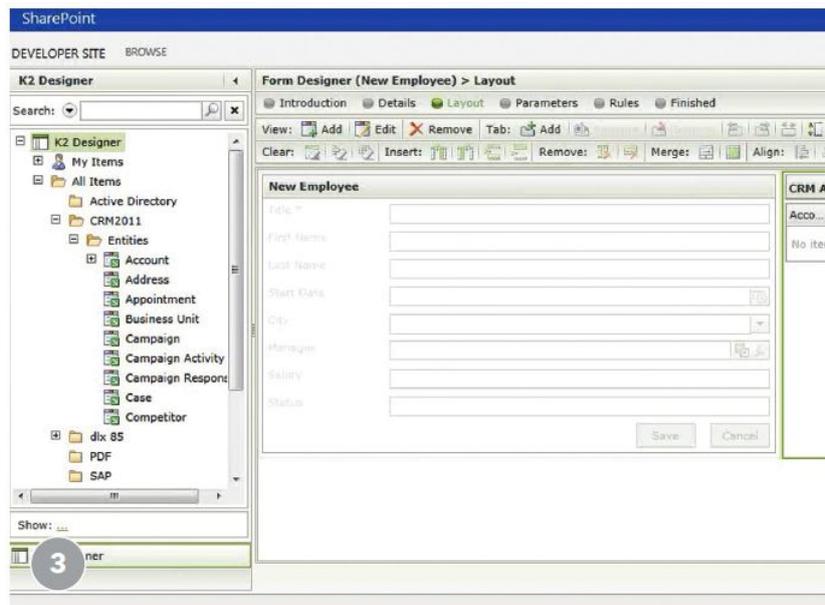


Abbildung 10: K2 Formular-Designer

Forms komplexe Formulare entwickelt und auch an externe Systeme angebunden werden können. Zusammen mit K2 Workflow lassen sich Formularbasierte Anwendungen vollständig in SharePoint integrieren.

### 2.4 Cit intelliForm

Die cit GmbH ist ein im Jahr 1993 gegründetes Unternehmen mit Hauptsitz in Dettlingen/Teck. Das Unternehmen hat sich auf rechnergestützte Lösungen von Informations- und Kommunikationstechnologien mit dem Branchenfokus Öffentliche Verwaltung und Kreditwirtschaft spezialisiert.<sup>9</sup>

Cit intelliForm ist eine ganzheitliche Lösung für ein Antrags- bzw. Fallmanagement. Abbildung 11 zeigt die Kernkomponenten der intelliForm Produktfamilie. Sie besteht im Wesentlichen aus den drei Komponenten: Input Management, Workflow Management und Output Management. In dieser Arbeit steht zunächst das Input Management im Vordergrund. Das Workflow Management wird im Abschnitt 4 näher betrachtet. Das Input Management steuert die Erfassung von Informationen mit Hilfe von Formularen. Mit Hilfe des intelliForm Composers lassen sich webbasierte Dialog-Assistenten erstellen, welche auf dem intelliForm Server, eine auf Java/JBOSS basierende Anwendungsplattform für das

<sup>9</sup> <http://www.cit.de/ueber-cit/unternehmensportrait>

gesamte elektronische Formular- und Fallmanagement, bereitgestellt werden.



Abbildung 11: Komponenten der intelliForm Produktfamilie [cit13a]

Dialog-Assistenten geben dem Benutzer eine klare Führung beim Ausfüllen eines Formulars. Der Dialog entspricht in etwa dem eines Installations-Assistenten (engl. Wizard), der den Benutzer mit Hilfe eines Dialoges durch das in Teilschritte zerlegte Formular führt. Die Dialog-Assistenten sind auf das Medium Web angepasst und mit angemessenen Hinweisen, Hilfetexten und Eingabeprüfungen versehen. Die Dialogsteuerung, Eingabeüberprüfung, Schnittstellen zu anderen Datenformaten (z. B. XML) sowie die Integration in Back-End Systeme werden von einer serverseitigen Runtime-Umgebung übernommen. Technisch wird es durch den Einsatz von Java-, Java Servlets- und Java Server Pages-Technologien umgesetzt und so zu einer mobilen Lösung. Zur Anzeige der Dialogführung auf dem Client (Webbrowser des Benutzers) sind keine weiteren aktiven Inhalte wie Java oder Javascript notwendig. Dies macht die Lösung browserunabhängig und offen für eine breite Zielgruppe.

Zur Erstellung solcher Formular-Assistenten gibt es zwei Möglichkeiten [cit13b, S. 1f]:

1. Eine Erstellung mit Hilfe des Software-Frameworks (intelliForm Assistant), bei der die einzelnen Dialogseiten als HTML-Dateien erstellt werden. Das Framework erweitert den Sprachumfang von HTML um weitere Elemente für Eingabefelder und Ablaufsteuerung in Form einer *JSP Tag Library*. Für diese Art der Erstellung von Formular-Assistenten werden HTML Kenntnisse und ein grundlegendes Verständnis für programmiertechnische Abläufe benötigt. [cit13b, S. 1f]
2. Eine Erstellung mit Hilfe des intelliForm Composers für Formular-Assistenten, ein Werkzeug zur grafischen Modellierung von Formular-Assistenten. Der Assistent kann ohne Programmierkenntnisse und Kenntnisse von HTML modelliert werden.

Da das Ziel dieser Arbeit das Herausstellen der Vorteile eines modellbasierten Ansatzes

ist, wird auf die erste Methode zur Erstellung eines Formular-Assistenten nicht näher eingegangen.

### 2.4.1 SemTalk

Der Composer setzt auf dem Prozessmodellierungs-Tool SemTalk auf. SemTalk ist das Produkt des Potsdamer Unternehmens Semtation GmbH<sup>10</sup> und ist seit über 10 Jahren in großen und mittleren Unternehmen im Einsatz. SemTalk selbst, setzt auf Microsoft Visio auf und erweitert dieses um eine XML-Datenbank. Neben einer SemTalk Lizenz ist demzufolge auch eine Microsoft Visio Lizenz notwendig, um es nutzen zu können. Zur begriffsbasierten Modellierung von Prozessen unterstützt SemTalk eine Reihe von verschiedenen Modellierungs-Notationen (unter anderem BPMN2.0, EPK) sowie weitere Möglichkeiten zur Erstellung von Organigrammen.

Abbildung 12 zeigt die Modellierungsoberfläche von SemTalk. Der Modellierer kann auf dem Zeichenblatt Shapes aus der Zeichenschablone per Drag & Drop platzieren und seinen Prozess abbilden. Unterstützt wird der Modellierer durch diverse Funktionen, die SemTalk bereitstellt. Der SemTalk-Explorer zeigt alle relevanten Diagramme, Objekte und Relationen an und über ihn lassen sich auch die Eigenschaften der Elemente ändern. Die BPMN-Toolstudie „Vergleich von BPMN-Tools zur Modellierung von Geschäftsprozessen“ ([Dec12]) zeigt SemTalk im Vergleich mit anderen Tools für die Notation BPMN und gibt weitere Eindrücke zum Modellieren mit SemTalk.

Was hat ein Prozessmodellierungs-Tool mit Formular-Assistenten zu tun? Die Modellierungsmethoden werden durch ein Metamodell beschrieben. Metamodelle sind Konzepte von Modellierungstechniken und legen fest, welche Modellelemente zu verwenden sind. Weiterhin definiert das Metamodell die Zusammenhänge zwischen den Elementen. In SemTalk lassen sich diese Metamodelle einfach anpassen. Dadurch ist SemTalk sehr flexibel in der Gestaltung der Modellierungsmethode.

---

<sup>10</sup> [www.semtalk.de](http://www.semtalk.de)

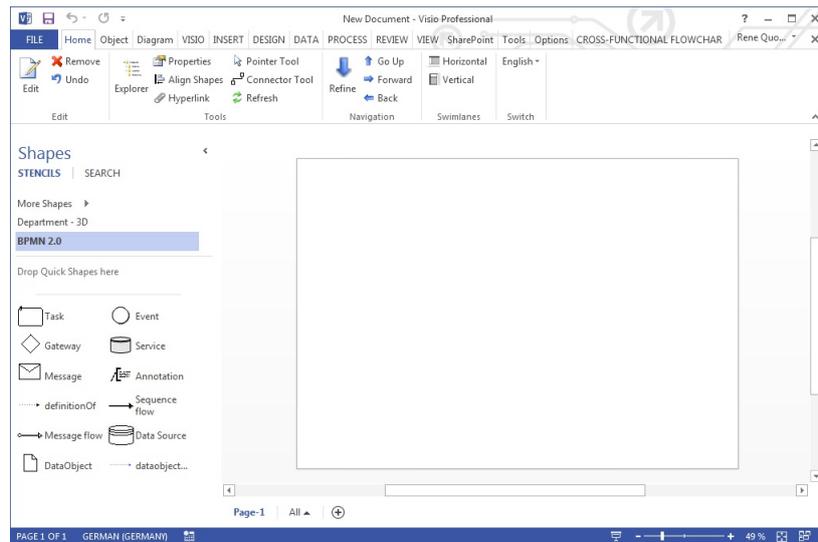


Abbildung 12: SemTalk Oberfläche

### 2.4.2 Cit intelliForm Composer für Formular-Assistenten

Der Composer ist, wie bereits erwähnt, ein Werkzeug zur grafischen Modellierung von Formular-Assistenten und setzt auf SemTalk auf. Abbildung 13 stellt die grobe Funktionsweise des Composers dar. Mit Hilfe von SemTalk erfolgt die semantische Modellierung eines Formularmodells. SemTalk selbst speichert das Modell in zwei Dateiformaten ab. Es wird im Visio-Dateiformat (\*.vsdx) und im XML-Format abgespeichert. Die XML-Datei enthält sämtliche Informationen des Modells. Der Composer liest das SemTalk-XML aus und erzeugt unter Verwendung einer speziellen JSP Tag Library die einzelnen Dateien des Formulars, welche in ein Webarchiv (WAR-Datei) gespeichert werden. Im Prinzip übernimmt der Composer das Erstellen der einzelnen JSP- bzw. XHTML-Seiten des Formulars, die auch per Hand mit Hilfe des intelliForm Assistants Framework erstellt werden könnten. Der generierte Formular-Assistent kann anschließend auf einem intelliForm Server bereitgestellt und ausgeführt werden.

Nachfolgend wird das Modellieren mit dem Composer kurz vorgestellt. Abbildung 14 zeigt den Startbildschirm des Composers. Da sich SemTalk fest in Microsoft Visio integrieren lässt, stehen sämtliche aus Visio bekannten Funktionalitäten zur Verfügung. Zusätzlich zu den Visio Funktionen erscheinen in der Ribbon weitere Reiter von SemTalk und des Composers. Der SemTalk Explorer (1) zeigt alle modellrelevanten Elemente. Eine speziell für den Composer entwickelte Schablone (2) enthält eine Auswahl von verschiedenen Shapes, mit denen Dialogseiten erstellt und Eingabefelder hinzugefügt werden. Sie

## 2 Grundlagen formularbasierter Anwendungen

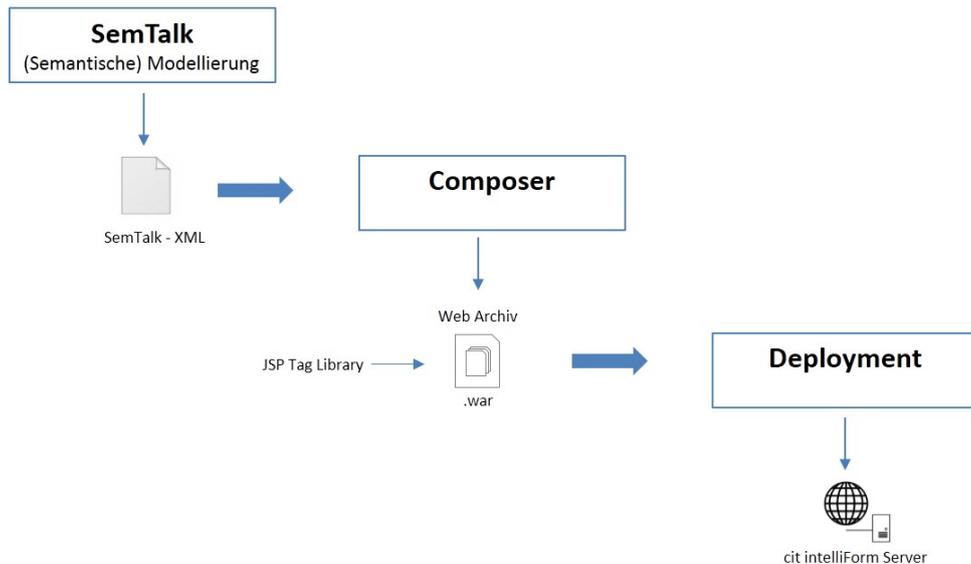


Abbildung 13: Formularerstellung mit dem cit intelliForm Composer für Formular-Assistenten

können einfach per Drag & Drop auf das Zeichenblatt gezogen werden.

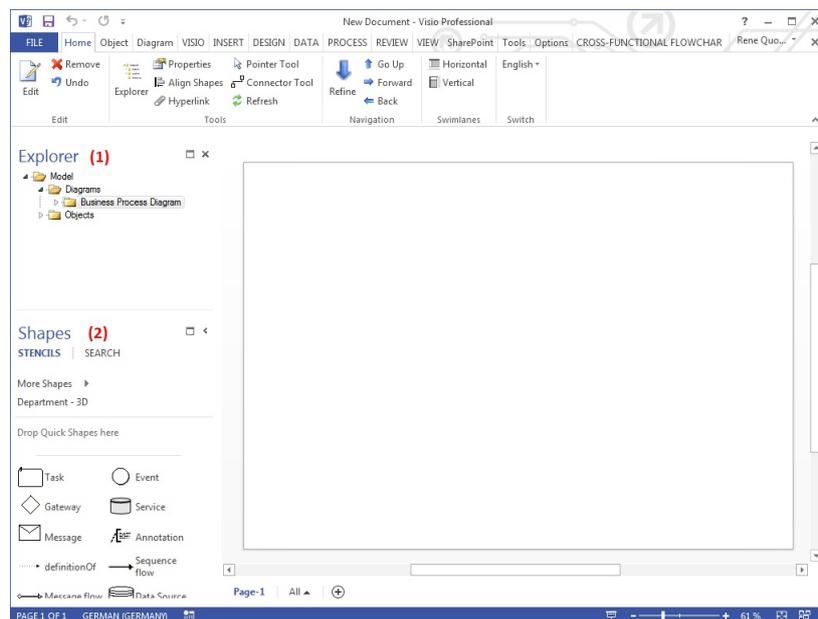


Abbildung 14: Composer - UI

Abbildung 15 zeigt ein Beispielformular. Das grüne Satz-Shape repräsentiert eine Dialogseite. Eine Dialogseite kann mit mehreren Formularfeldern oder weiteren Sätzen durch Relationen verbunden werden. Die Formularfelder werden durch die hellblauen Shapes dargestellt. Die Schablone enthält verschiedene Typen von Eingabefeldern zur Darstellung der einzelnen Feldtypen. Mit Hilfe der unterschiedlichen Feldtypen ist der Composer in der Lage, beim Erstellen des Formular-Assistenten Feldvalidierungen zu implemen-

## 2 Grundlagen formularbasierter Anwendungen

tieren. Das heißt, in einem Zahlenfeld werden zum Beispiel im fertigen Assistenten nur Zahlen als gültige Eingabe akzeptiert. Eine entsprechende Regel muss der Modellierer nicht selbst anlegen.

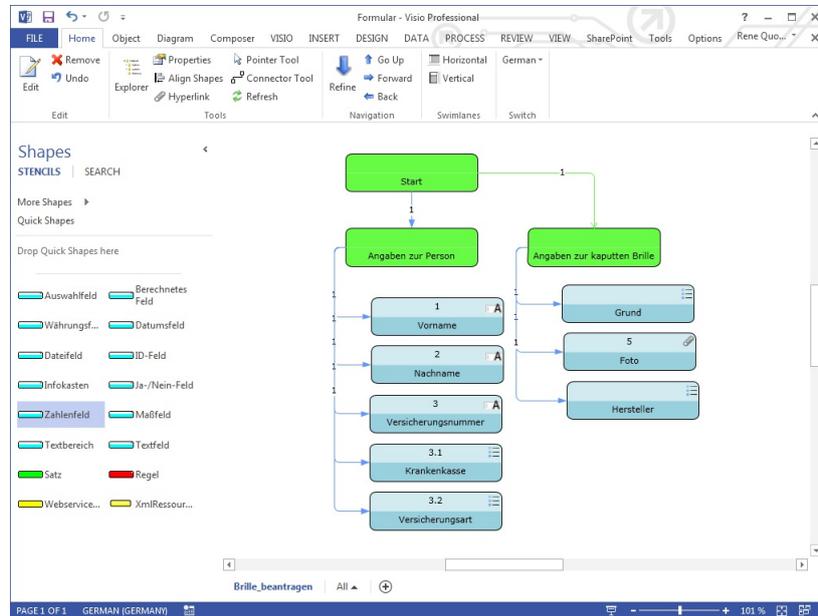


Abbildung 15: Composer - Beispiel

Weitere Funktionalitäten wie zum Beispiel Überschriften, Zusatztexte, Feldbeschriftungen und dessen Position, Reihenfolge der Formularfelder und Wertelisten für Auswahlfelder werden mit Hilfe von Attributen an den jeweiligen Shapes oder aber an einer Relation gesteuert. Abbildung 16 zeigt die Attributliste eines Satz-Shapes.

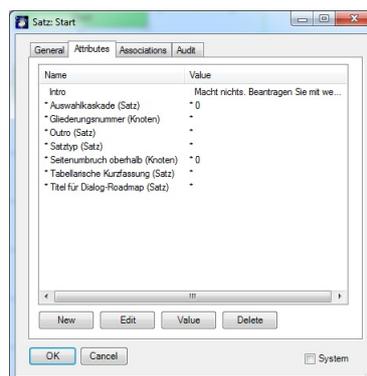


Abbildung 16: Attribute (Satz)

Neben den Shapes für Formularfelder gibt es Shapes für externe Ressourcen. Diese dienen dazu, Daten aus externen Quellen (z.B. einer XML-Datei) im Formular zu verwenden bzw. im Falle des Webservices auch Daten an einem Webservice zu senden. Das Regel-Shape dient zur Prüfung von Abhängigkeiten zwischen zwei oder mehreren

Eingabefeldern.

Durch Starten des Composers wird aus dem Modell ein Formular-Assistent generiert (siehe Abbildung 17). Das Web-Archiv kann anschließend auf einem intelliForm Server bereitgestellt werden. Die Daten des ausgefüllten Formulars werden wie bei den XForms in einem XML-Datensatz serialisiert und können zum Beispiel von einem angebundenen Fallmanagementsystem weiter verarbeitet werden.

Abbildung 17: Beispiel des fertigen Formular-Assistenten

Mit dem Composer schlägt die cit eine andere Richtung in der Erstellung von Formularen ein. Das Alleinstellungsmerkmal dieses Ansatzes besteht in der grafischen Modellierung eines Formular-Assistenten. Der Ansatz erlaubt dadurch eine fachliche Trennung vom Inhalt des Formulars und dessen Darstellung. Im Formulardesign werden lediglich die zum Formular gehörenden Dialogseiten und Formularfelder sowie dessen Reihenfolge und Abhängigkeiten voneinander modelliert. Die Gestaltung des Formulars wird hingegen mit Hilfe von Vorlagen für den Composer serverseitig gesteuert. So entstehen aus mehreren Formulardesigns verschiedene einheitlich gestaltete Formular-Assistenten.

### 2.5 Microsoft Info Path und Forms Services

Microsoft stellt mit InfoPath ein Werkzeug zum Erstellen von XML-basierten Formularen bereit. Mit InfoPath können sowohl Ausgabemasken zur Anzeige von Werten aus einer Datenbank als auch Eingabemasken zum Füllen von Datenbanken erstellt werden. Es können nicht nur Eingabe- und Ausgabemasken erstellt werden, sondern auch mehrere Formulare zusammengeführt und zum Beispiel mit einer Pivot-Tabelle ausgewertet werden. Auch der Import von Word- oder Excel-Formularen ist möglich. Seit der Version InfoPath 2007 wird zwischen dem InfoPath Designer und dem InfoPath Filler unterschieden. Der InfoPath Designer dient zum Erstellen dynamischer Formulare. InfoPath Filler

## 2 Grundlagen formularbasierter Anwendungen

---

ist eine Clientanwendung zum Ausfüllen der im InfoPath Designer erstellten Formulare.

Der InfoPath Designer unterscheidet sich vom cit Composer insofern, dass in InfoPath Formulare nach dem „WYSIWYG“-Prinzip erstellt werden. Das Formular wird in InfoPath auf einem Zeichenblatt genauso entworfen, wie es später beim Ausfüllen aussieht. Abbildung 18 zeigt die Benutzeroberfläche von InfoPath. Wie in Microsoft Word und Excel stehen in InfoPath Funktionen zum Formatieren von Texten, Einfügen von Tabellen und ganzen Farbschemas bereit. Diese können einfach über die Ribbon eingefügt werden. Die einzelnen Steuerelemente, die InfoPath zur Verfügung stellt, können per Drag & Drop auf dem Zeichenblatt positioniert werden. Alle Steuerelemente werden in einer Baumstruktur (ähnlich dem SemTalk Explorer) aufgelistet.

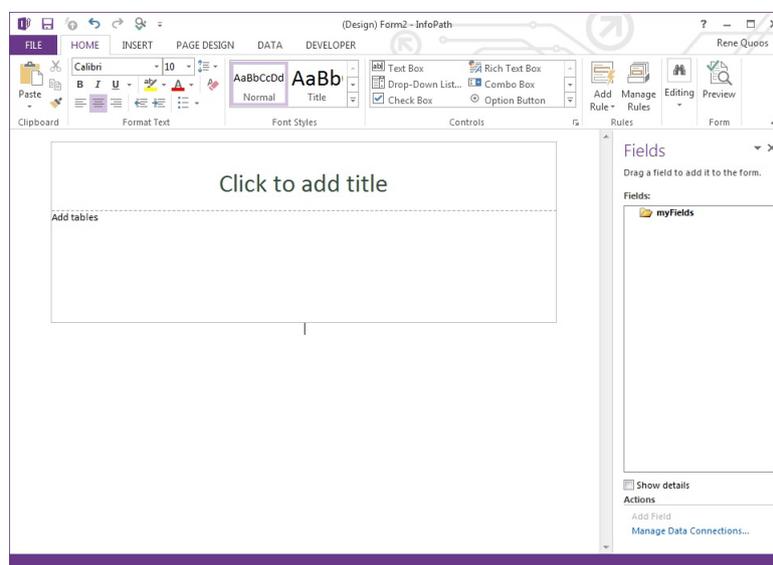


Abbildung 18: InfoPath Designer - Oberfläche

Feldvalidierungen, das Ein- oder Ausblenden von Steuerelementen, der Ansichtswchsel und vieles mehr können über sogenannte Regeln (engl. rules) gesteuert werden. Regeln stellen einen wesentlichen Bestandteil eines Formulars dar, durch sie wird das Formular interaktiv. Weiterhin bietet InfoPath eine Reihe von Schnittstellen zu externen Daten an. So lassen sich zum Beispiel die Werte für Auswahlfelder aus SharePoint Listen oder XML-Dateien nutzen. Neben dem Heranschaffen von Daten existieren Schnittstellen zum Versenden des Formulars. Dazu gehört der Versand als E-Mail, das Ablegen in SharePoint oder das weiterreichen an einem Webservice.

Mit InfoPath können zwei Arten von Formularen erstellt werden. Es lassen sich InfoPath Filler und browserfähige Formulare mit InfoPath anfertigen. Für erstere wird immer die Client-Anwendung zum Ausfüllen benötigt. Die browserfähigen Formulare haben den

## 2 Grundlagen formularbasierter Anwendungen

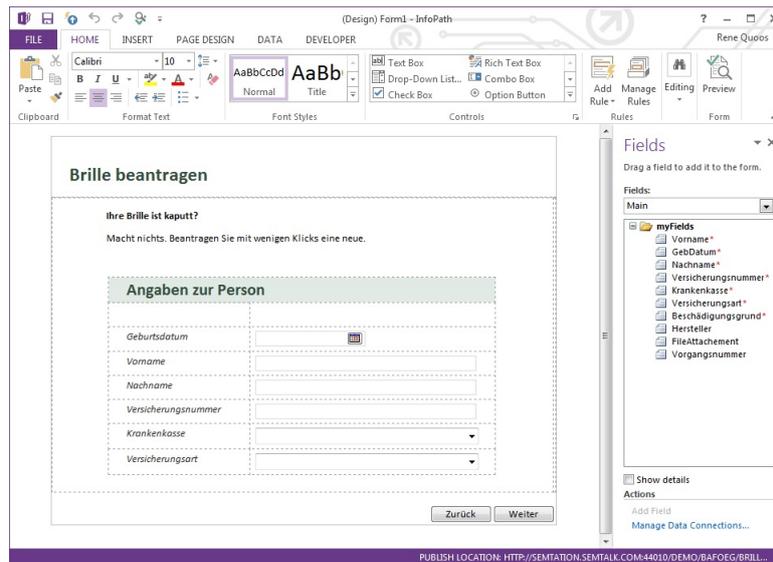


Abbildung 19: InfoPath Designer - Formular

Vorteil, dass sie sich auch in einem Web-Browser ausführen lassen. Allerdings lassen sich InfoPath Webformulare nicht in jeder beliebigen Webanwendung ausführen. Dies funktioniert nur, unter Verwendung von Microsoft Technologien. Abbildung 20 zeigt welche Technologien eine Rolle spielen.

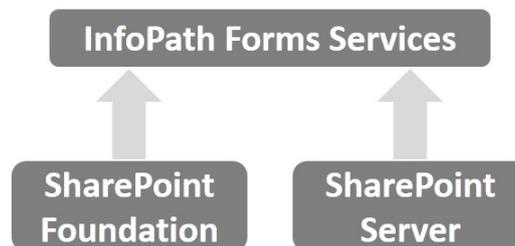


Abbildung 20: Forms Services

Für die Ausführung der Formulare im Browser werden die Forms Services benötigt. Forms Services stellen eine Servertechnologie basierend auf SharePoint Foundation dar und enthalten Funktionen zur Verwaltung von Formularvorlagen an einem zentralen Ort und zum Ausfüllen eines Formulars im Browser. Die Forms Services werden durch den SharePoint Server oder der SharePoint Foundation (bis Version 2007 Windows SharePoint Services) bereitgestellt. Microsoft SharePoint ist eine Plattform, die verschiedene Funktionalitäten für Unternehmen bereitstellt. SharePoint wird in vielen unterschiedlichen Einsatzszenarien eingesetzt, z. B. als Unternehmens-, Informations- oder Kooperationsportal oder aber als zentrale Dokumentenverwaltung. Die SharePoint Foundation ist eine Webanwendung und stellt Portalseiten zur kollaborativen Arbeit bereit. Sie enthält

Schnittstellen zu anderen Microsoft Produkten wie InfoPath, Access oder Outlook. Der SharePoint Server erweitert die Funktionalität der SharePoint Foundation. Die SharePoint Foundation ist eine kostengünstige Alternative für kleinere Unternehmen. Der SharePoint Server ist ein Portal für Webseiten und untergeordneten Webseiten, welche mit Webparts eingerichtet werden. Die Enterprise Version vom SharePoint enthält zusätzliche Business Intelligence Funktionen, eine erweiterte Suchfunktion sowie die Nutzung von Workflows. Beide Technologien, SharePoint Foundation und SharePoint Server, stellen die Forms Services bereit.

## 3 Vergleich des modellbasierten Ansatzes mit InfoPath

Im vorherigen Abschnitt wurden zwei Ansätze zur Formularerstellung vorgestellt. Dieser Abschnitt betrachtet die vorgestellten Lösungen cit intelliForm und InfoPath detailliert. Eine detaillierte Betrachtung ist notwendig, um eine Aussage treffen zu können, ob ein modellbasierter Ansatz einen tatsächlichen Mehrwert gegenüber der WYSIWYG-Erstellung mit sich bringt. Um diese Vorteile herauszuarbeiten, werden die beiden Anwendungen miteinander verglichen. Der Vergleich erfolgt anhand eines Anwendungsszenarios. Das Anwendungsszenario beschreibt ein bereits vorhandenes und praxisnahes Formular, welches bereits mit cit intelliForm umgesetzt wurde. Anhand des Formularmodells wird ein entsprechendes InfoPath Formular entworfen. Es gilt, dabei die Gemeinsamkeiten und Unterschiede beim Erstellen der Formulare aufzuzeigen und miteinander zu vergleichen.

Für den Vergleich wurde das Bundesausbildungsförderungsgesetz (BAföG)<sup>11</sup> ausgewählt. Um eine Ausbildungsförderung zu erhalten, muss ein entsprechender Antrag gestellt werden. Für diesen Antrag gibt es ein Formular, bestehend aus mehreren Formblättern. Das BAföG-Antragsformular ist ein sehr großes und aufgrund von Abhängigkeiten einzelner Felder bzw. ganzer Formblätter auch ein sehr komplexes Beispiel. Das Land Baden-Württemberg stellt einen Online-Assistenten zum Ausfüllen des Formulars auf ihrer Webseite bereit<sup>12</sup>. Dieser Assistent wurde mit Hilfe von cit intelliForm realisiert. Die Umsetzung des BAföG-Antragsformulars mit InfoPath wurde auf einen kleineren Ausschnitt beschränkt. Der gewählte Ausschnitt ist ausreichend, um die wesentlichen Punkte beider Ansätze miteinander zu vergleichen und Unterschiede feststellen zu können.

Im Anhang ist der in InfoPath umgesetzte Ausschnitt des cit Modells (Anhang Seite XIV und XV) sowie das InfoPath Formular (Anhang Seite XVI bis XXII) abgebildet.

---

<sup>11</sup> Das BAföG regelt die Ausbildungsförderung für Jugendliche und junge Erwachsene in Deutschland. Weitere Informationen erhalten Sie unter: <http://www.bafoeg.bmbf.de>

<sup>12</sup> <https://ekp.dvwbw.de/intelliform/forms/bafoeg-bw/bw/eantrag/index>

#### 3.1 Dialogseiten und Ansichten

Dialogseiten und Ansichten stellen die einzelnen Schritte im Ausfüllprozess des Formulars dar. Sie enthalten Info-Texte, Formularfelder, die durch den Benutzer auszufüllen sind, und Schaltflächen, um zur nächsten bzw. vorherigen Seite zu gelangen.

Die Dialogseiten werden im Composer in der Regel durch das grüne Satz-Shape repräsentiert. Der Satz kann mit mehreren Eingabefeldern und weiteren Sätzen durch Relationen verbunden werden. Jedes cit Modell hat genau einen Satz, der keine eingehende Relation hat. Er wird als Start-Satz bezeichnet und ist die erste Dialogseite des Formular-Assistenten. Ein Satz stellt nicht in allen Fällen eine eigene Dialogseite dar. Soll ein Satz und dessen zugeordneten Eingabefelder auf der gleichen Seite des übergeordneten Satzes angezeigt werden, so kann an der Relation zum Satz ein Attribut „Typ“ gesetzt werden. Es gibt zwei Typen von Relationen. Die Assoziation ist standardmäßig eingestellt, wenn zwei Elemente miteinander verbunden werden. Wird der Typ auf den Wert „Composition“ gesetzt, bedeutet das für den Composer, dass der Satz nicht auf einer eigenen Dialogseite angezeigt wird. Stattdessen wird er als Block auf dem übergeordneten Satz angezeigt. Der Satz „Familienstand“ aus Abb. 21 ist eine Composition.

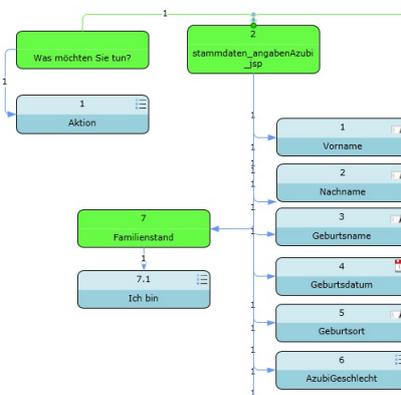


Abbildung 21: Composer - Sätze und Eingabefelder

In InfoPath werden die Formularseiten als Ansichten bezeichnet. Der große Unterschied zwischen InfoPath und dem Composer ist, dass in InfoPath die Ansichten nicht abstrahiert abgebildet werden. Das Ergebnis der Ansicht entspricht genau dem, was auf dem Zeichenblatt im Design-Modus zu sehen ist (siehe Abb. 22 (1)).

Dies bedeutet, dass der Formular-Ersteller sich Gedanken über das Layout macht und die Elemente entsprechend anordnen muss. Das Wechseln und Anzeigen von vorhandenen Ansichten geschieht über die Ribbon im Tab „Page Design“ (Abb. 22 (2)). Im cit Modell

### 3 Vergleich des modellbasierten Ansatzes mit InfoPath

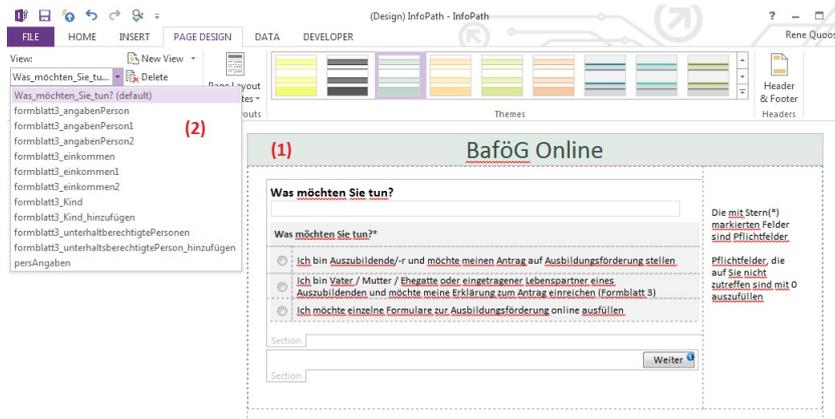


Abbildung 22: InfoPath - Ansichten

wird die Darstellung teilweise über Attribute an Relationen und an den Shapes selbst gesteuert. Dazu gehören unter anderem die Position der Beschriftung, das Einfügen von Seitenumbrüchen oder Ein- bzw. Ausleitungstexte (Intro und Outro).

#### 3.1.1 Eingabefelder und Steuerelemente

Eingabefelder und Steuerelemente stellen das Herzstück eines Formulars dar. Im cit Modell enthält die Schablone alle verwendbaren Eingabefelder (siehe Abb. 23). Mit diesen Eingabefeldern sind einige Voreinstellungen zur Feldvalidierung verbunden. In einem Textfeld können Werte vom Typ „string“ (string steht für sämtliche Zeichen) durch den Benutzer eingegeben werden. Infokästen dienen der Anzeige von Informationen in Form von Text. Die Eingaben bei einem Zahlenfeld sind auf Zahlen beschränkt. Das Zahlenfeld dient zur Eingabe von „reinen“ Zahlen. Für Angaben mit Einheiten (wie z.B. Gramm (g), Kilogramm (kg) usw.) wird das Maßfeld und für Währungen das Währungsfeld verwendet.

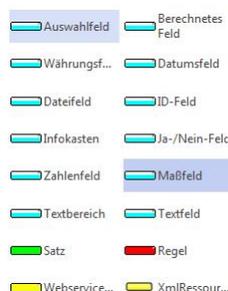


Abbildung 23: Composer - Eingabefelder

Über die Eigenschaften der Eingabefelder können Wertelisten für Auswahlfelder und Formatierungen angegeben werden. Der Composer stellt dem Modellierer Eingabefelder

### 3 Vergleich des modellbasierten Ansatzes mit InfoPath

---

für verschiedene Einsatzzwecke bereit. Schaltflächen um zur nächsten bzw. vorherigen Seiten zu gelangen müssen nicht angelegt werden, sondern ergeben sich aus den Relationen.

In InfoPath gibt es ähnliche Steuerelemente bzw. Eingabefelder. Abbildung 24 zeigt die vorhandenen Elemente. Ein Textfeld repräsentiert ein normales Eingabefeld für Eingaben durch den Benutzer. In den Eigenschaften des Textfeldes (25) kann ein Datentyp festgelegt werden. Der Datentyp beschränkt die Eingabewerte auf den ausgewählten Typ. Weiterhin lassen sich im Eigenschaften-Dialog sämtlicher Steuerelemente in InfoPath die Darstellung und das Verhalten anpassen. Dazu gehören unter anderem auch Bildschirmhinweise oder alternative Zugriffstasten, die die Zugänglichkeit erhöhen. Bei Auswahlfeldern kann in den Steuerelement-Eigenschaften ein für den Benutzer nicht sichtbarer interner Wert sowie ein Anzeigewert für die Auswahl festgelegt werden.

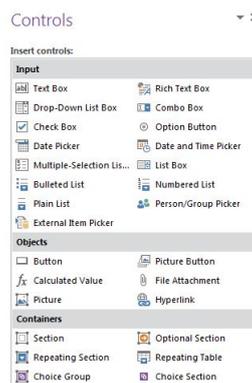


Abbildung 24: InfoPath - Steuerelemente

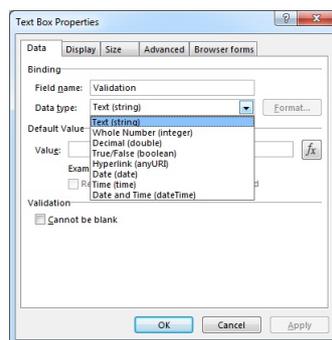


Abbildung 25: InfoPath - Eigenschaften des Textfeldes

Insgesamt bietet InfoPath eine größere Auswahl verschiedener Steuerelemente. Dies bedeutet jedoch nicht, dass der Composer eine geringere Anzahl von Eingabefeldern besitzt. Im Composer werden, im Gegensatz zu InfoPath, viele Eigenschaften oder Feldtypen über Attribute am Shape bzw. an der Relation gesteuert. Für ein Auswahlfeld in InfoPath

existieren drei Steuerelemente (Drop-Down-List Box, Combo-Box und Option-Button). Der Composer bildet diese Felder mit einem Shape, dem Auswahlfeld, ab. Der Typ des Auswahlfeldes (ob Option-Button oder Drop-Down-Liste) wird über ein Attribut gesteuert.

#### 3.1.2 Beschriftungen

Beschriftungen und Überschriften sind notwendig, um den Benutzer zu sagen, welche Informationen in den Eingabefeldern der Dialogseite eingegeben werden müssen. Das Formularmodell steuert diese Überschriften, Beschriftungen und Hinweistexte über Attribute am Shape (siehe Abb. 26). Die Überschrift einer Dialogseite wird am Satz-Shape mit dem Titel Attribut festgelegt. Für die Beschriftung an Eingabefeldern sind die Attribute „*Beschriftung*“ und „*Beschriftungsposition*“ verantwortlich. Ersteres stellt den Anzeigetext der Beschriftung dar und letztere bestimmt die Position der Beschriftung. Zur Position der Beschriftung gibt es drei Auswahlmöglichkeiten. „*Prefix*“ stellt die Beschriftung dem Eingabefeld voran, „*Suffix*“ stellt die Beschriftung dem Eingabefeld nach und „*Question*“ setzt die Beschriftung über das Eingabefeld.

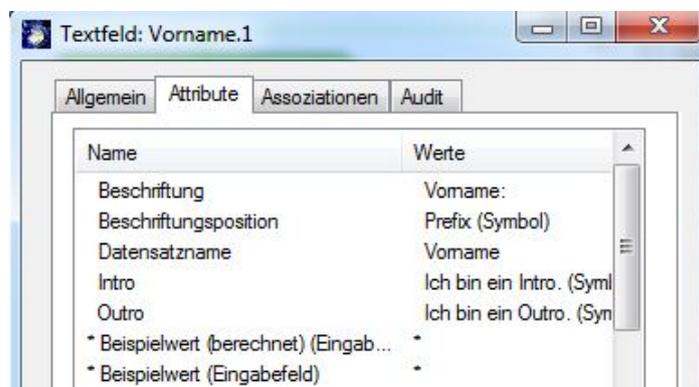


Abbildung 26: Composer - Beschriftungen

Zusätzlich zu der Beschriftung kann am Satz bzw. am Eingabefeld ein Einleitungs- bzw. Ausleitungstext (*Intro* und *Outro*) angegeben werden. Der Einleitungstext (*Intro*) wird über dem Eingabefeld und dessen Beschriftung platziert und der Ausleitungstext (*Outro*) unter dem Eingabefeld. Zur Platzierung von weiteren Texten auf der Dialogseite dient das *Infokasten*-Shape. Die Position des Textes wird durch die Gliederungsnummer bestimmt.

Anders als im Modell werden im InfoPath die Überschriften, Beschriftungen und Texte direkt an die gewünschte Stelle, wo sie im Formular auftauchen sollen, geschrieben. Abbildung 27 zeigt den Aufbau einer Ansicht des BAföG-Formulars mit InfoPath. Die Texte

für die Auswahlmöglichkeiten des Auswahlfeldes werden wie Überschriften auf dem Zeichenblatt platziert. Den einzelnen Optionen des Auswahlfeldes kann ein interner Wert zugeordnet werden. Im Datensatz ist ausschließlich der intern vergebene Wert vorhanden.

The screenshot shows a web form titled 'Bafög Online' with a section for 'Einkommen'. The form contains a text box for 'also die des Kalenderjahres: \*' and a radio button selection for 'Art der Erwerbstätigkeit: \*'. The options are: 'erwerbstätig als rentenversicherungsspflichtige/r Arbeitnehmer/in (z.B. Arbeiter/in, Angestellte/r) oder in Ausbildung', 'erwerbstätig als nichtrentenversicherungsspflichtige/r Arbeitnehmer/in (z.B. Beamtin/Beamtin oder Beamtin/Beamtin im Ruhestand)', 'erwerbstätig als Nichtarbeitnehmer/in (z.B. Selbstständige/r) oder auf Antrag von der Rentenversicherungspflicht befreite/r oder wegen geringfügiger Beschäftigung versicherungsfrei/r Arbeitnehmer/in', and 'nicht erwerbstätig oder: im Ruhestandsalter nicht erwerbstätig (z.B. Altersrentner/in)'. There are also 'Zurück' and 'Weiter' buttons at the bottom.

Abbildung 27: InfoPath - Überschriften und Beschriftungen

#### 3.1.3 Unterformulare und wiederholende Abschnitte

Im Verlaufe des BAföG-Antrages müssen Angaben zu eventuell vorhandenen Kindern bzw. unterhaltsberechtigten Personen gemacht werden. Bei mehreren Kindern oder unterhaltsberechtigten Personen müssen pro Kind bzw. pro Person eine bestimmte Anzahl von Angaben gemacht werden. Im cit Formular wird dies durch eine besondere Art und Weise gelöst. In unserem Beispiel erzeugt der Composer für die Angaben zum Kind ein Unterformular. Ein Unterformular ist ein Formular innerhalb eines Formulars. Wird auf der entsprechenden Dialogseite im Formular auf die Frage: Haben Sie Kinder? mit „Ja“ geantwortet, wird die Dialogseite aus Abbildung 28 als nächster Schritt im Ausfüllprozess angezeigt. Abbildung 28 zeigt eine Verteilerseite mit einer Auflistung von bereits eingetragenen Kindern. Weiterhin erzeugt der Composer Schaltflächen zum Ändern und Öffnen bereits vorhandener Einträge bzw. zum Hinzufügen neuer Einträge. Beim Klick auf die Schaltfläche „Kind hinzufügen“ öffnet sich das Unterformular für die Angaben zum Kind (Abbildung 29). Für jedes Kind wird ein eigenes Unterformular erzeugt.

Der entsprechende Ausschnitt aus dem Modell ist in Abbildung 30 zu sehen. Der Satz „formblatt3\_angabenPerson2\_jsp“ enthält die Frage, ob Kinder vorhanden sind. Ist dies der Fall, müssen auf der nächsten Dialogseite Angaben zum Kind getätigt werden. Es folgt zunächst eine Übersichtsseite mit einer tabellarischen Auflistung aller Kinder. Das Unterformular selbst setzt sich aus den Sätzen „Kind“ und „Kind Ausbildung“ zusammen.

### 3 Vergleich des modellbasierten Ansatzes mit InfoPath

Abbildung 28: Composer - Unterformular 1

Abbildung 29: Composer - Unterformular 2

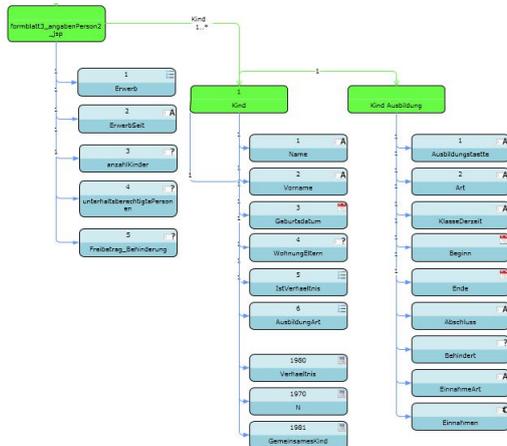


Abbildung 30: Composer - Modell des Unterformulars

Besonders interessant für solche Unterformulare ist die Relation, die den Satz „formblatt3\_angabenPerson2\_jsp“ mit dem Satz „Kind“ verbindet. Sie ist deswegen so interessant, weil anhand der gesetzten Attribute dieser Relation eine Verteilerseite zum Hinzufügen mehrerer Einträge generiert wird. Ausschlaggebend für diese Eigenschaft ist das Attribut „Multiplizität“ (siehe Abb. 31). Die Multiplizität legt fest, wie viele Listeneinträge gemacht werden müssen. In diesem Beispiel ist der Wert „1..\*“, d. h. es muss mindestens ein Eintrag gemacht werden. Mit dem Attribut „Maximale Anzahl“ kann eine Obergrenze von maximal möglichen Einträgen gesetzt werden. Überschriften bzw. Ein- und Ausleitungstexte der Verteilerseite werden über die Attribute der Relation gesteuert. Das Attribut „Tabellarische Kurzfassung“ steuert die Anzeige der Listeneinträge auf der Verteilerseite.

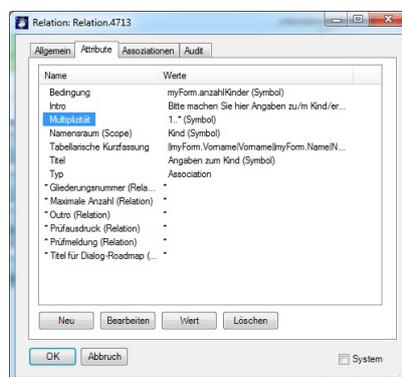


Abbildung 31: Composer - Multiplizität und tabellarische Kurzfassung

InfoPath bietet ähnliche Funktionen zum Abbilden von mehrfach auszufüllenden Abschnitten. Um ein Mehrfachausfüllen einer Menge von Feldern zu ermöglichen, muss das Steuerelement „Repeating Section“ verwendet werden. Es wird wie üblich per Drag & Drop auf dem Zeichenblatt platziert. Anschließend können dem Steuerelement weitere Steuerelemente (z.B. Eingabefelder) hinzugefügt werden. Alle Steuerelemente, die sich innerhalb der „Repeating Section“ befinden, werden beim Ausfüllen des Formulars zum mehrfachen Ausfüllen kopiert. Abbildung 32 zeigt einen Ausschnitt des InfoPath-Formulars im Design-Modus.

Abbildung 33 zeigt das Aussehen des sich wiederholenden Abschnittes im fertigen InfoPath Formular. Standardmäßig wird ein Satz von Eingabefeldern beim Ausfüllen des Formulars angezeigt. Um weitere Kinder hinzuzufügen, gibt es für den Benutzer zwei Möglichkeiten dies zu tun, entweder über per Klick auf den Text „Kind hinzufügen“ (1) oder über ein Kontextmenü, welches sich am linken oberen Rand (2) öffnen lässt. Mit Hilfe des Kontextmenüs hat der Benutzer neben dem Hinzufügen noch die Möglichkeit einen

### 3 Vergleich des modellbasierten Ansatzes mit InfoPath

The screenshot shows the 'BaföG Online' form in Design Mode. The main title is 'BaföG Online'. Below it, the section is titled 'Angaben zum Kind: Allgemeines'. The form contains several fields: 'Name:' (text box), 'Vorname:' (text box), 'Geburtsdatum:' (calendar icon), 'Wohnung bei den Eltern:' (radio buttons for 'Ja' and 'Nein'), and 'Verhältnis:' (radio buttons for 'Gemeinsames Kind der Eltern...', 'Kind nur im Verhältnis zum Vater...', 'Kind nur im Verhältnis zur Mutter...', 'Kind nur im Verhältnis zum Ehegatten...', and 'Stiefkind, Pflegekind, enkelkind soweit in Ihrem Haushalt aufgenommen'). Below these is 'Das Kind:' (radio buttons for 'geht zur Schule/studiert/befindet sich in Ausbildung' and 'ist nicht in Ausbildung aber unterhaltsberechtig (auch nicht schulpflichtige Kinder)'). At the bottom, there is a 'Repeating Section' indicator, 'Abbrechen' and 'Weiter' buttons, and a 'Section' label.

Abbildung 32: InfoPath - Repeating Section - Design Modus

Eintrag wieder zu löschen.

The screenshot shows the 'BaföG Online' form in Preview Mode. The main title is 'BaföG Online'. Below it, the section is titled 'Angaben zum Kind: Allgemeines'. The form contains several fields: 'Name:' (text box), 'Vorname:' (text box), 'Geburtsdatum:' (calendar icon), 'Wohnung bei den Eltern:' (radio buttons for 'Ja' and 'Nein'), and 'Verhältnis:' (radio buttons for 'Gemeinsames Kind der Eltern...', 'Kind nur im Verhältnis zum Vater...', 'Kind nur im Verhältnis zur Mutter...', 'Kind nur im Verhältnis zum Ehegatten...', and 'Stiefkind, Pflegekind, enkelkind soweit in Ihrem Haushalt aufgenommen'). Below these is 'Das Kind:' (radio buttons for 'geht zur Schule/studiert/befindet sich in Ausbildung' and 'ist nicht in Ausbildung aber unterhaltsberechtig (auch nicht schulpflichtige Kinder)'). At the bottom, there is a 'Repeating Section' indicator with a plus sign and the text 'Kind hinzufügen', 'Abbrechen' and 'Weiter' buttons, and a 'Section' label. Red annotations are present: a '(2)' next to the section title and a '(1)' next to the 'Kind hinzufügen' button.

Abbildung 33: InfoPath - Repeating Section - Preview Modus

Anders als im cit Formular werden die Mehrfacheinträge nicht in separaten Unterformularen ausgefüllt, sondern untereinander in einer Ansicht. Das „Nachbauen“ einer Verteilerseite, wie in der cit Version, ist möglich. Dabei ist zu beachten, dass die einzelnen Seiten und das Anzeigen der Einträge aus den wiederholenden Abschnitten manuell erstellt werden müssen und weiterführende Kenntnisse in XPath erfordern.

### 3.1.4 Reihenfolge und Ablaufsteuerung

Im cit Modell bildet der Start-Satz die erste Dialogseite des Formular-Assistenten. Alle weiteren Sätze, die direkt mit dem Start-Satz verbunden sind, werden als Folgesätze bezeichnet. Folgesätze können weitere Sätze (Untersätze) besitzen. Die Reihenfolge der Sätze bzw. der Eingabefelder einer Dialogseite wird hauptsächlich über ein Attribut am jeweiligen Objekt gesteuert. Mit Hilfe des Attributs „Gliederungsnummer“ können die Sätze in die gewünschte Reihenfolge gebracht werden. Für den Fall, dass keine Gliederungsnummern vergeben werden, erfolgt die Bestimmung der Reihenfolge anhand einer alphabetischen Sortierung nach dem Objektnamen.

In Abbildung 34 ist der Start-Satz „Was möchten Sie tun“ sowie der Folgesatz „stammdaten\_angabenAzubi\_jsp“ und dessen Untersätze „Familienstand“ und „formblatt3\_angabenPerson\_jsp“ abgebildet. Die Gliederungsnummern der Objekte sind durch eine Zahl oberhalb des Objektname im Shape gekennzeichnet.

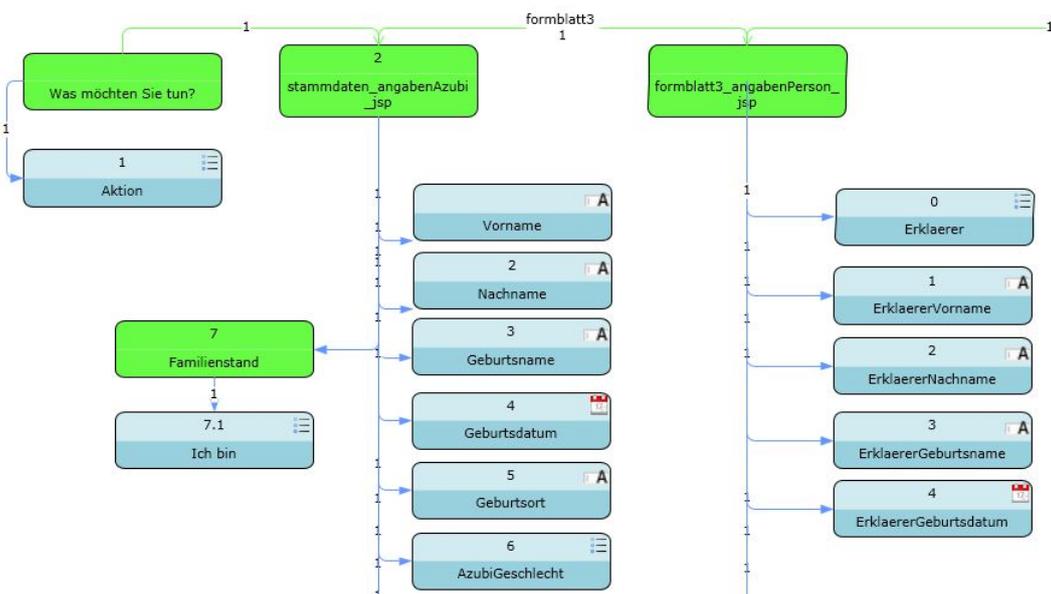


Abbildung 34: Composer - Reihenfolge und Ablaufsteuerung

Die Abfolge der einzelnen Dialogseiten erfolgt nach dem Schema: Start-Satz, Folge-Satz A, Untersatz A1, Untersatz A2, Folge-Satz B, Untersatz B1, Untersatz B2.

Im fertigen Dialogassistenten sieht die Reihenfolge der Dialogseiten wie folgt aus:

1. Was möchten Sie tun?
2. stammdaten\_angabenAzubi\_jsp

#### 3. formblatt3\_angabenPerson\_jsp

Es lässt sich eine beliebig tiefe Schachtelung der Sätze erzeugen. Im fertigen Formularassistenten ist die Schachtelung für den Benutzer nicht zu erkennen. Er erhält weiterhin eine lineare Abfolge von Dialogseiten. Die Schaltflächen, um zur jeweils nächsten bzw. vorherigen Dialogseite zu gelangen, werden automatisch erstellt.

Die Reihenfolge der einzelnen Eingabefelder eines Satzes wird ebenfalls durch die Gliederungsnummer oder durch eine alphabetische Sortierung bestimmt. Der Satz „Familienstand“ stellt in diesem Fall eine Besonderheit dar. Dieser Satz wurde als „Composition“ modelliert und als Block auf der übergeordneten Dialogseite platziert. Er erscheint an siebenter Stelle, nach dem Feld „AzubiGeschlecht“, auf der Dialogseite des Satzes „stammdaten\_angabenAzubi\_jsp“.

Das Attribut Gliederungsnummer kann auch an einer Relation vergeben werden. Das ist zum Beispiel dann der Fall, wenn ein ganzer Satz bzw. ein Eingabefeld auf mehreren Dialogseiten verwendet wird und der Satz bzw. das Feld an einer bestimmten Position, wie zum Beispiel an Position 3 auf Seite A und Position 4 auf Seite B, platziert werden soll. In diesem Fall ist die Zuordnung der Gliederungsnummer an der Relation erforderlich.

InfoPath bietet im eigentlichen Sinne keine wirkliche Ablaufsteuerung. Der Benutzer kann im InfoPath Formular über die Ribbon die Ansicht wechseln, sofern diese Option nicht deaktiviert wurde. Abbildung 35 zeigt das Kontextmenü zur Navigation zwischen den Ansichten im Vorschau-Modus des Formulars.



Abbildung 35: InfoPath - Ansichten Wechseln über die Ribbon

Eine weitere Möglichkeit, Ansichten zu wechseln, lässt sich über Schaltflächen realisieren. Mit Hilfe von sogenannten *Regeln* können an den Schaltflächen Aktionen zum Wechseln der Ansicht hinterlegt werden. Für eine lineare Abfolge der Ansichten müsste auf

jeder Ansicht eine Schaltfläche mit einem Wechsel zur jeweils nächsten Ansicht erstellt werden.

#### 3.1.5 Feldvalidierungen, Bedingungen und Regeln

Eingaben durch den Benutzer und Ausgaben in einem Formular müssen überprüft werden. Die Überprüfung von Benutzereingaben ist ein wesentlicher Bestandteil von Webformularen. Durch die Feldvalidierung können Fehleingaben erkannt und dem Benutzer mitgeteilt werden, an welcher Stelle im Formular eine Fehleingabe erfolgte (Abb. 36). Neben der Feldvalidierung spielen Bedingungen und Regeln eine weitere Rolle im Formular. Bedingungen und Regeln sorgen für ein dynamisches Verhalten des Formulars, wie zum Beispiel das Ein- oder Ausblenden von ein oder mehreren Eingabefeldern oder gesamter Dialogseiten.

Ministerium für Wissenschaft, Forschung und Kunst  
Baden-Württemberg

**Bundesausbildungsförderungsgesetz (BAföG)**  
Personenbezogene Angaben zum Auszubildenden

**Fehler beim Ausfüllen: Bitte überprüfen Sie Ihre Angaben.**

Bitte ausfüllen.  
Vorname:\*

Bitte ausfüllen.  
Nachname:\*

Geburtsname:

Bitte ausfüllen.  
Geburtsdatum:\*

Bitte ausfüllen.  
Geburtsort:\*

Abbrechen Unterbrechen... Zurück Weiter

Die mit Stern (\*) markierten Felder sind Pflichtfelder.  
Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

[Nutzungsbedingungen](#) | [Erklärung zum Datenschutz](#) | [Impressum](#)

Abbildung 36: Eingabeüberprüfung

In cit intelliForm wird zwischen der Feldvalidierung und Seitenvalidierung unterschieden. Die Feldvalidierung dient zur Überprüfung des einzelnen Wertes und erfolgt in mehreren Stufen (Pflichtfeldprüfung, typspezifische Formatprüfung, Prüfung mit regulären Ausdrücken und Prüfung mit Javascript). Entsprechende Feldvalidierungen können mit Hilfe von Attributen am Eingabefeld-Shape bzw. an der Relation gesetzt werden. In Abbildung 37 sind die dafür zuständigen Felder zu sehen. Die typspezifische Formatprüfung ist abhängig vom verwendeten Eingabefeld. Eingaben in Zahlenfelder, Maßfelder, Währungs-

felder und Datumsfelder werden automatisch auf ein fest hinterlegtes Format überprüft (z. B. Zahlenfelder enthalten nur Zahlen).

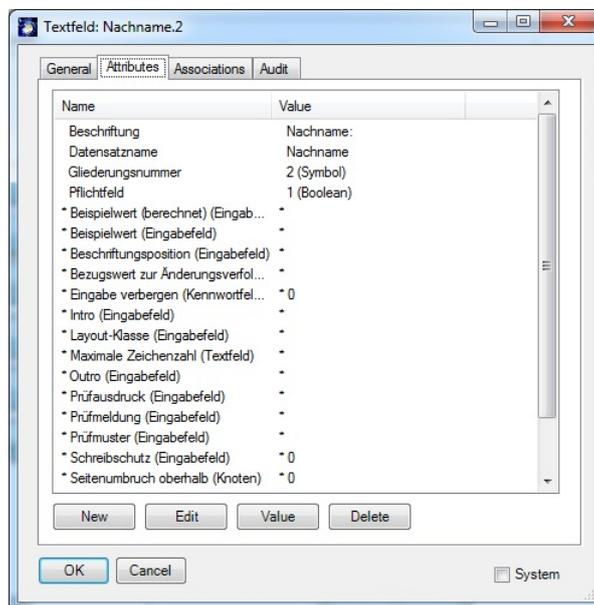


Abbildung 37: Composer - Feldvalidierung - Attribute am Eingabefeld

Felder, die ausgefüllt werden müssen, lassen sich mit dem Attribut „Pflichtfeld“ markieren. Reguläre Ausdrücke<sup>13</sup> dienen zur Überprüfung der Eingaben anhand eines bestimmten Musters. Im Attribut „Prüfausdruck“ können komplexere Prüfungen mit Javascript vorgenommen werden. Mit dem Attribut „Bezugswert zur Prüfaufforderung“ lässt sich dem Benutzer ein Hinweis anzeigen, die Eingaben eines Feldes auf Richtigkeit zu überprüfen. Dies ist genau dann notwendig, wenn ein Eingabefeld von einem anderen abhängig ist. Ein Beispiel: In einem unternehmensinternen Formular müssen auf einer Dialogseite die Abteilung und auf einer weiteren die Kontaktdaten, die Telefonnummer usw., eingegeben werden. Die verschiedenen Abteilungen haben unterschiedliche Telefonnummern (z. B. aufgrund von verschiedenen Standorten). Beim einmaligen Durchlauf des Formulars füllt der Benutzer das Formular korrekt aus. Fehler treten erst dann auf, wenn er nach dem ersten Ausfüllen auf die Seite zur Angabe der Abteilung „zurückblättert“ und die Abteilung ändert. Damit würde sich möglicherweise auch die Telefonnummer auf der Kontaktseite ändern. Im Formular steht allerdings noch die zuvor eingegebene Telefonnummer. Mit Hilfe des Attributs „Bezugswert zur Prüfaufforderung“ am Feld der Telefonnummer kann festgelegt werden, dass wenn sich der Wert im Feld „Abteilung“ ändert der Benutzer einen Hinweis zur Überprüfung der Telefonnummer angezeigt bekommt.

<sup>13</sup> kurze Erläuterung von regulären Ausdrücken

### 3 Vergleich des modellbasierten Ansatzes mit InfoPath

Bedingungen werden im cit Modell an den Relationen gesetzt. Durch Bedingungen wird gesteuert, ob Eingabefelder oder ganze Sätze im Formular angezeigt werden oder nicht. Ein Beispiel aus dem BAföG-Modell ist die Angabe zu den Kindern. Die dazugehörigen Dialogseiten für die Angaben zu Kindern müssen nur ausgefüllt werden, wenn zuvor die Frage nach Kindern mit „Ja“ beantwortet wurde.

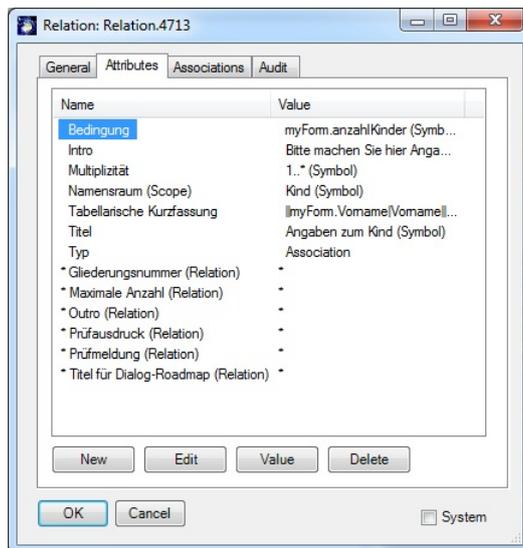


Abbildung 38: Composer - Bedingungen an Relationen

Die Seitenvalidierung dient zur Überprüfung logischer Abhängigkeiten mehrerer Felder einer Dialogseite. So muss das Feld „verheiratet mit“ nur dann ausgefüllt werden, wenn zuvor als Familienstand „verheiratet“ angegeben wurde. Die grafische Darstellung solcher Seitenvalidierungen erfolgt mit dem Regel-Shape (siehe Abb. 39). Eine Regel hat die Attribute „Prüfausdruck“, „Prüfmeldung“ und „Gliederungsnummer“.

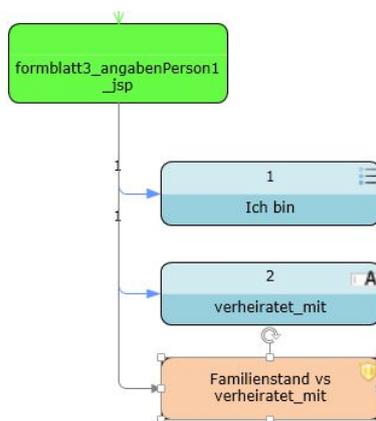


Abbildung 39: Composer - Seitenvalidierung mit Regeln

Sowohl die Feld- als auch die Seitenvalidierung werden vom intelliForm Server ausgeführt. Sie erfolgt nach Klick auf die „Weiter“-Schaltfläche. Scheitern einzelne Validierung

gen, wird die entsprechende Dialogseite mit zusätzlichen Hinweistexten erneut angezeigt. Die Hinweistexte lassen sich durch den Modellierer im Attribut „Prüfmeldung“ definieren.

In InfoPath werden Feldvalidierungen, Abhängigkeiten und logische Zusammenhänge einzelner Felder über Regeln gesteuert. InfoPath Regeln sind das Pendant zu den Attributen Bedingung, Prüfausdruck, Prüfmuster und dem Regel-Shape im cit Modell. Ähnlich wie bei cit wird abhängig von Datentyp eines Feldes automatisch eine typspezifische Formatprüfung vorgenommen. Alles, was über die typspezifische Prüfung hinausgeht, muss der Formularersteller mit Hilfe von Regeln festlegen.

Der Datentyp und das Markieren eines Feldes als Pflichtfeld bzw. Read-Only lassen sich sowohl über die Steuerelement-Eigenschaften als auch über die Ribbon einstellen (Abbildung 40). Das Erzwingen eines Pflichtfeldes lässt sich auch mit einer Regel (dazu später mehr) erzeugen.

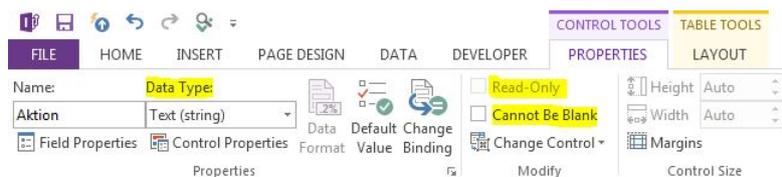


Abbildung 40: InfoPath - Steuerelement-Eigenschaften (Ribbon)

Feldvalidierungen allein sind in einem Webformular nicht ausreichend. Es muss dem ausfüllenden Benutzer kenntlich gemacht werden, an welcher Stelle im Formular eine Fehleingabe stattgefunden hat. Im cit Modell wurde dazu zum Beispiel das Attribut „Pflichtfeld“ gesetzt. Bei Fehleingaben werden entsprechende Hinweistexte an den jeweiligen Feldern angezeigt. Die Validierung erfolgt nach dem Klick auf die „Weiter“-Schaltfläche und lässt bei fehlerhafter Prüfung kein Voranschreiten zur nächsten Dialogseite zu.

Um ein entsprechendes Verhalten im InfoPath zu erzeugen, sind einige Dinge zu beachten. Als erstes muss in InfoPath die Abfolge der einzelnen Ansichten eingestellt werden. Standardmäßig lassen sich die Ansichten über die Ribbon auswählen. Um den Benutzer Schritt für Schritt durch die einzelnen Ansichten des Formulars zu führen, muss der Ansichtenwechsel über die Ribbon deaktiviert werden. Dies geschieht über die Formular-Optionen. Da nun keine Möglichkeit besteht über das Ribbon Menü die Ansichten zu wechseln, müssen Schaltflächen auf allen Ansichten eingefügt werden. Regeln an der Schaltfläche steuern, welche Ansicht als nächstes angezeigt werden soll. Abbildung 41

### 3 Vergleich des modellbasierten Ansatzes mit InfoPath

zeigt die erste Dialogseite des BAföG-Formulars. Die erste Dialogseite enthält das Auswahlfeld „Aktion“, welches ein Pflichtfeld ist und ein Textfeld zur Anzeige eines Hinweistextes. Mit der Schaltfläche „Weiter“ soll der Benutzer zur nächsten Dialogseite im Ausfüllprozess gelangen. Sie enthält bereits mehrere Regeln um den Ansichtenwechsel zu realisieren. Die ausgewählte Regel („goNext“) dient dem eigentlichen Ansichtenwechsel. Mit ihr wird festgelegt, welche Ansicht als nächstes angezeigt wird und unter welcher Bedingung dies geschehen soll. Die Bedingung „das Feld Aktion darf nicht leer sein“ ist deshalb notwendig, um zu verhindern, dass der Ansichtenwechsel bei einem nicht ausgefülltem Pflichtfeld durchgeführt wird. Es ist zwar eine entsprechende Pflichtfeld-Regel am Feld Aktion, allerdings würde ohne die Bedingung an der Schaltfläche der Ansichtenwechsel trotzdem stattfinden. Das Absenden des Formulars am Ende des Ausfüllprozesses würde aufgrund der Pflichtfeld-Markierung scheitern.

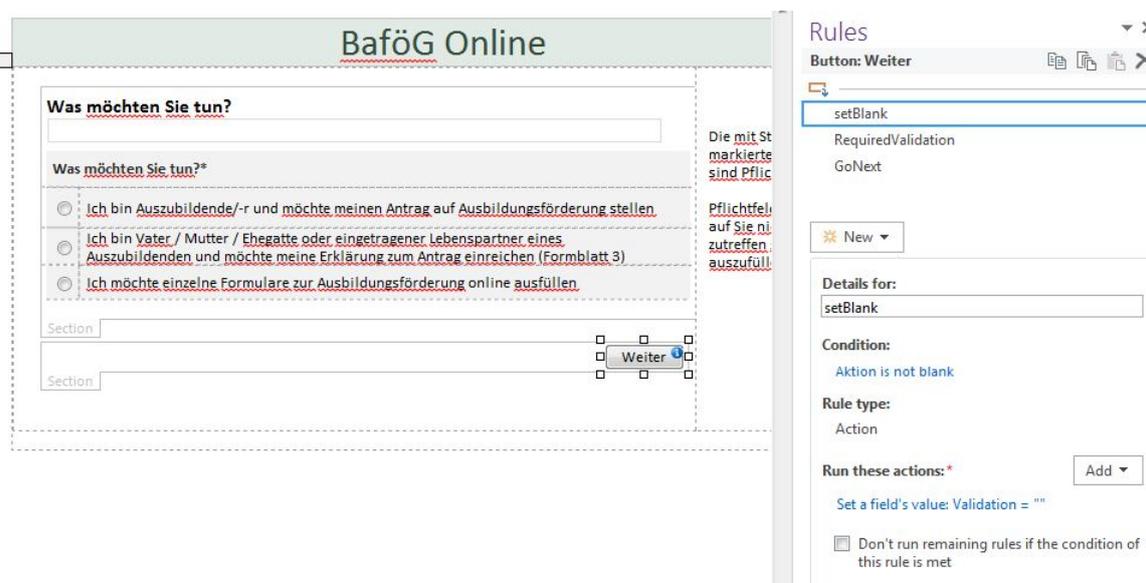


Abbildung 41: InfoPath - Ansichtenwechsel

Für die logische Abfolge der einzelnen Ansichten müssen entsprechende Bedingungen an der Weiter-Schaltfläche gesetzt werden. Im Verlaufe des BAföG-Antrags muss angegeben werden, ob Kinder vorhanden sind oder nicht. Wird die Frage mit „Ja“ beantwortet, folgen zunächst Ansichten, auf denen Angaben zu den Kindern gemacht werden müssen. Der Weiter-Button auf der Ansicht mit der Frage nach Kindern muss demzufolge zwei Regeln für den Ansichtenwechsel enthalten. Eine Regel, die auf die Ansicht des Unterformulars, mit der Bedingung die Frage wurde mit „Ja“ beantwortet, weist und eine weitere zur nächsten logischen Ansicht des Formulars für den Fall das die Frage mit „Nein“ beantwortet wurde.

### 3 Vergleich des modellbasierten Ansatzes mit InfoPath

Zur Anzeige von Hinweistexten für Falscheingaben, zum Beispiel bei leer gelassenen Pflichtfeldern, werden weitere Regeln und Steuerelemente benötigt. Für die Ausgabe des Hinweistextes wurde eine TextBox unterhalb der Überschrift der Seite platziert. Diese TextBox enthält zwei Regeln zur Formatierung (siehe Abbildung 42). Sie wird zunächst nicht im Formular angezeigt. Die zweite Regel formatiert die TextBox mit einer roten Hintergrundfarbe und fettgedruckten Buchstaben. Die TextBox sollte als Read-Only deklariert werden, um eine Interaktion durch den Benutzer zu unterbinden.

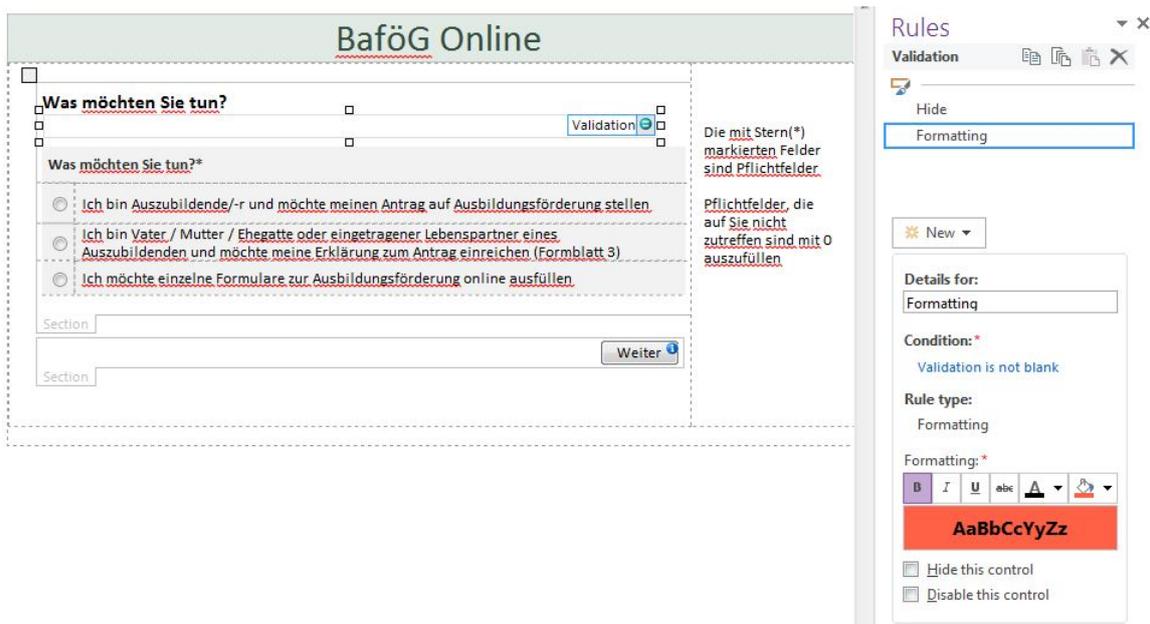


Abbildung 42: InfoPath - Feldvalidierung

Zusätzlich zu den zwei Regeln an der TextBox kommen zwei weitere Regeln an der Schaltfläche hinzu. Die Regeln „setBlank“ und „RequiredValidation“ aus Abbildung 41 steuern das Anzeigen des Hinweistextes. Beim Klick auf die Schaltfläche werden die Regeln in folgender Reihenfolge ausgeführt: 1. setBlank, 2. RequiredValidation, 3. goNext. Die erste Regel setzt den Wert der TextBox für den Hinweistext zurück, denn die TextBox soll nur angezeigt werden, wenn diese leer ist. Die Regel „RequiredValidation“ füllt die TextBox mit einem Hinweistext, sofern keine Auswahl im Auswahlfeld getroffen wurde. Dadurch wird beim ersten Klick auf „Weiter“ ohne das Treffen einer Auswahl des Auswahlfeldes, der entsprechende Hinweistext im Formular angezeigt und der Ansichtenwechsel nicht vollzogen. Trifft der Benutzer nun eine Auswahl und klickt auf „Weiter“ wird der Hinweistext entfernt und die TextBox ist leer. Da nun die TextBox leer ist und das Pflichtfeld einen Wert enthält, wird die Aktion der Regel „RequiredValidation“ nicht ausgeführt und folglich zur nächsten Ansicht gewechselt.

### 3 Vergleich des modellbasierten Ansatzes mit InfoPath

Dieses Konstrukt an Regeln und Steuerelementen entspricht in etwa dem Verhalten des cit Formulars. Eine Pflichtfeld-Markierung des Eingabefeldes könnte in diesem Fall sogar weggelassen werden, da der Benutzer aufgrund der Regeln am Weiter-Button keine Möglichkeit hat, bei Falscheingaben und fehlenden Eingaben zur nächsten Ansicht zu gelangen.

Ein weiterer wichtiger Aspekt der Webformulare interaktiv macht, ist das Ein- bzw. Ausblenden von Eingabefeldern oder ganzer Ansichten. Sollen ganze Ansichten in Abhängigkeit eines anderen Feldes nicht im Ausfüllprozess angezeigt werden, geschieht dies wie bereits gezeigt, über Regeln an den Schaltflächen. Beim Ein- oder Ausblenden eines oder mehrerer Eingabefelder treten weitere Schwierigkeiten auf. Abbildung 43 zeigt die zweite Ansicht des BAföG-Antrags im Design-Modus.

The screenshot shows the 'Bafög Online' form in design mode. The form is titled 'Bafög Online' and is divided into sections. The first section is 'Personenbezogene Angaben zum Auszubildenden'. It contains a table with the following fields:

Label	Input Field	Notes
Vorname:	<input type="text"/>	Die mit Stern(*) markierten Felder sind Pflichtfelder. Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen
Nachname:	<input type="text"/>	
Geburtsname:	<input type="text"/>	
Geburtsdatum:	<input type="text"/> <input type="button" value="📅"/>	
Geburtsort:	<input type="text"/>	
Section		
Geschlecht:	<input type="radio"/> weiblich <input type="radio"/> männlich	
Section		
<b>Familienstand</b>		
Ich bin: *	<input type="text"/>	
<input type="radio"/> ledig		
<input type="radio"/> verheiratet/in eingetragener Lebenspartnerschaft		
<input type="radio"/> dauernd getrennt lebend		
<input type="radio"/> verwitwet		
<input type="radio"/> geschieden		
Section		
Section		
Section		
Zurück Weiter		

Abbildung 43: InfoPath - Ein- und Ausblenden von Eingabefelder I

Für den Aufbau des Formulars wurde eine Tabelle genutzt. Die erste Spalte der Tabelle enthält in der Regel die Beschriftung der Steuerelemente, die zweite Spalte das Eingabefeld und das Feld zur Anzeige von Hinweistexten. Eingabefelder lassen sich mit einer einfachen Formatierungsregel ausblenden (Abb. refimg:vgl23).

Bei dem verwendeten Design tritt allerdings folgendes Problem auf: Das Eingabefeld wird

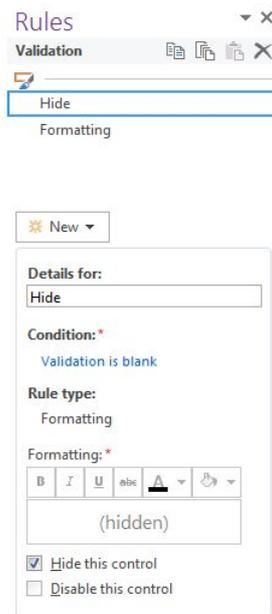


Abbildung 44: InfoPath - Ein- und Ausblenden von Eingabefelder II

korrekt ausgeblendet, die Beschriftung jedoch nicht. Sie stellt lediglich normalen Text im Formular dar. Daher ist es notwendig, eine oder mehrere Zeilen der Tabelle auszublenden. Das Ausblenden einer ganzen Zeile der Tabelle ist nicht möglich. Soll ein Feld bzw. in diesem Falle immer eine ganze Tabellenzeile ausgeblendet werden, müssen sogenannte „Sections“ verwendet werden. Innerhalb der Section können Steuerelemente platziert werden. Anstatt die Regel für das Ausblenden am entsprechenden Eingabefeld zu erstellen, wird die Regel an der Section angewandt. In dem Beispiel aus Abbildung 43 existiert eine entsprechende Regel für die mittlere Section. Damit der Antragsteller im späteren Formular nicht merkt, dass eine Zeile ausgeblendet wird, müssen innerhalb der Section die Tabellen entsprechend formatiert werden. Das bedeutet der Formular-Designer muss Seitenabstände, Spaltenbreiten und -höhen gegebenenfalls anpassen, um einen nahtlosen Übergang zu gewährleisten.

Sind Felder innerhalb der Section als Pflichtfelder markiert, so ist dies beim Erstellen der Regeln für den Ansichtenwechsel zu berücksichtigen. Eine Pflichtfeldprüfung allein reicht nicht mehr aus, denn wenn der Benutzer das Pflichtfeld nicht ausfüllen kann, weil es nicht angezeigt wird, gelangt er nicht zur nächsten Ansicht. Dies muss mit zusätzlichen Bedingungen und weiteren Regeln abgefangen werden.

Technisch gesehen erzeugt der cit Composer für den Fall, dass einzelne Felder oder ganze Feldgruppen (Composition) eines Satzes ausgeblendet werden, jeweils eine ei-

genständige Dialogseite. Für InfoPath bedeutet dieses Verhalten ein Mehrfachmodellieren der Ansichten für die einzelnen Fälle.

## 3.2 Modularisierung

Große Formulare mit vielen Dialogseiten und Steuerelementen können schnell unübersichtlich werden. Modularisierung erlaubt es, übersichtliche, modulare und überschaubare Modelle zu erstellen. Der Composer bietet mehrere Techniken zur Modularisierung. Es kann vorkommen, dass in einem Formular bestimmte Felder mehrfach verwendet werden. Dies ist zum Beispiel der Fall, wenn in einem Bestellformular eine Rechnungsadresse und eine Lieferadresse erfasst werden soll. Unter Verwendung von Namensräumen an den Relationen lassen sich einzelne Eingabefelder oder ganze Sätze mehrfach verwenden. Anstatt die Felder für Straße, PLZ und Ort doppelt zu modellieren (mit unterschiedlichen Datensatznamen, weil sonst ein Namenskonflikt im XML-Datensatz entstehen würde) können die gemeinsamen Eingabefelder wie in Abbildung 45 dargestellt in einem Satz Adresse zusammengefasst werden. Um den Satz „Adresse“ sowohl für die Lieferanschrift als auch für die Rechnungsanschrift zu verwenden, ist an den Relationen zur Adresse ein Namensraum zu vergeben. Durch Angabe des Namensraums verändert sich der XML-Datensatz und löst den entstandenen Namenskonflikt auf.

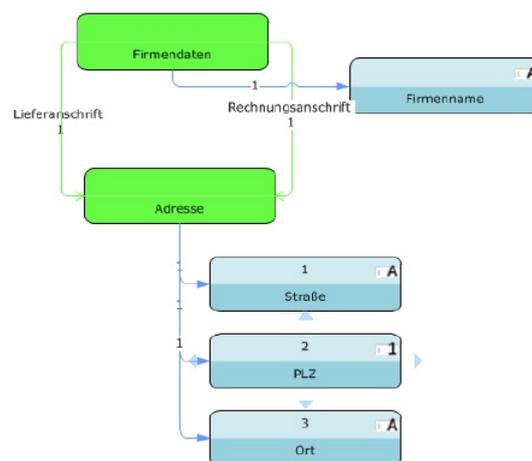


Abbildung 45: Composer - Mehrfachverwendung von Sätzen und Eingabefelder [cit13c, S. 103]

Eine weitere Möglichkeit zur Modularisierung großer Modelle ist die Auslagerung von Shapes auf ein anderes Zeichenblatt bzw. in externe Dateien. Diese Technik wird in Sem-Talk als Verfeinerung bezeichnet. Verfeinerungen eignen sich, um mehrere Shapes auf

dem führenden Zeichenblatt zu einem Shape zusammen zu fassen. Damit wird eine gewisse Übersichtlichkeit gewahrt, während die Details auf einem anderen Zeichenblatt dargestellt werden. Weisen viele Modelle Gemeinsamkeiten auf, so ist es sinnvoll diese Gemeinsamkeiten zur Vereinheitlichung in einer externen Datei zu modellieren.

Eine weitere Möglichkeit zur Modularisierung im Composer ist das Arbeiten mit Formularbausteinen. Formularbausteine sind Composer-Modelle, die sich als Komponenten in anderen Modellen verwenden lassen. Sie werden wie ein normales Formularmodell mit einem Startsatz und nachgeordneten Feldern und Sätzen modelliert. Ein Satz „Anschrift“ mit den Feldern „Straße“, „PLZ“ und „Ort“ ließe sich einfach in anderen Modellen einfügen. Eine Besonderheit der Formularbausteine sind die Bausteinvarianten. Varianten sind kontextspezifische Erweiterungen der Bausteine. Für die Lieferanschrift soll zusätzlich eine Telefonnummer angezeigt werden. Der Baustein wird mit Hilfe der Variante „Lieferanschrift“ um das Feld „Telefonnummer“ erweitert. Der Modellierer kann nun je nach Kontext den Baustein einfügen.

InfoPath bietet ebenfalls Möglichkeiten zur Modularisierung. Ähnlich wie im Composer lassen sich in InfoPath sogenannte „Template Parts“ erstellen. Template Parts lassen sich wie Formularvorlagen in InfoPath auf einem Zeichenblatt erstellen. Die Template Parts sind mit den Bausteinen vom Composer zu vergleichen. Sie sind Ausschnitte aus einer Formularvorlage, die sich in mehreren Formularen verwenden lassen. Sie können Steuerelemente, Datenressourcen, Datenverbindungen, Feldvalidierungen und Regeln enthalten. Freigegebene Template Parts werden als Steuerelemente ins Formular eingefügt (siehe Abbildung 46).

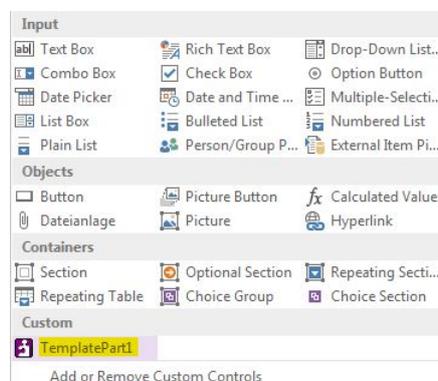


Abbildung 46: Composer - Mehrfachverwendung von Sätzen und Eingabefelder

Die Funktionen in Template Parts unterliegen einigen Beschränkungen. Zu den Einschränkungen gehören unter anderem bestimmte Ansichtsfunktionen, die Verwendung

von Skripten und von Managed Code sowie bestimmte Funktionen in Bezug auf die Formularvorlage als Ganzes (Digitale Signaturen, Sicherheitseinstellungen, benutzerdefinierte Verhaltensweisen beim Öffnen und Speichern des Formulars etc.).<sup>14</sup>

#### **3.3 Externe Ressourcen und Datenverbindungen**

Externe Ressourcen werden häufig in Zusammenhang mit Auswahlfeldern verwendet. Anstatt die Werteliste für ein Auswahlfeld direkt im Formular einzugeben, lassen sich die Werte aus externen Dateien (z. B. .xml-Dateien) verwenden. Dies hat den Vorteil, dass Werteliste an einem zentralen Ort verwaltet werden können und in allen Formularen auf dieselben Ressourcen zugreifen können. Im cit Formular lassen sich solche externe Ressourcen (XML-Dateien) mit dem XmlRessourcen-Shape einbinden. In den Attributen wird eine URL auf die entsprechende XML-Ressource gesetzt. Mit Angabe eines XPath's lässt sich die Werteliste für ein Auswahlfeld filtern. Eine weitere Möglichkeit ist die Anbindung von Webservices. Webservices stellen Funktionen dar, die von einem anderen System bereitgestellt werden. Solche Dienste lassen sich ebenfalls im Formular verwenden, zum Beispiel zur Berechnung des BAföG-Betrags.

In InfoPath gibt es Datenverbindungen. Datenverbindungen dienen zum einen zur Integration von externen Daten aus Webservices, SharePoint Listen, XML-Dateien oder Datenbanken und zum anderen zum Absenden des Formulars an eine E-Mail Adresse, zu einer SharePoint Bibliothek oder zu einem Webservice. Ein Assistent unterstützt den Formularersteller bei der Integration der Datenverbindungen. Daten aus externen XML-Dateien lassen sich nach dem Hinzufügen der Datenverbindung einfach als Werteliste eines Auswahlfeldes nutzen. Abbildung 47 zeigt die Einstellungen für die Datenverbindung. Als Data Source wird eine Ressourcen-Datei ausgewählt, die die Daten hält. In diesem Fall eine XML-Datei namens „characters“. Im nächsten Feld können Einträge für die Werteliste aus der XML-Datei gefiltert und ausgewählt werden.

---

<sup>14</sup> [Mic13c]

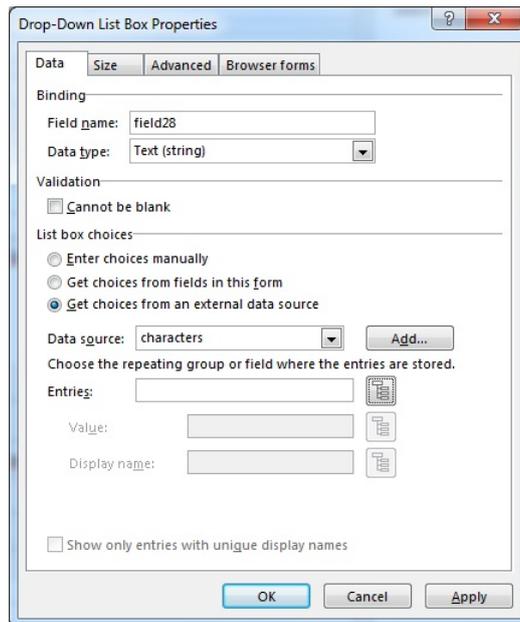


Abbildung 47: InfoPath - Datenverbindungen

#### 3.4 Ausführbarer Programmcode

Ausführbarer Programmcode spielt für IntelliForm eine wesentliche Rolle. Der Composer selbst erstellt aus dem Formularmodell \*.jsp-Seiten (Java Server Pages), die auf dem IntelliForm Server ausgeführt werden. Java Server Pages haben den Vorteil, dass die Java basierten Formulare in jedem Webbrowser nutzbar sind und ohne clientseitig ausgeführten Programmcode auskommt. Die Mehrzahl an Attributen von Sätzen, Eingabefelder und Relationen beinhalten Javascript Ausdrücke. Diese können zum einen einfache Ausdrücke sein, wie z. B. als Bedingung „myForm.Aktion != 'erklaren\_familie'“ aber auch komplexere Funktionen enthalten. Javascript Funktionen lassen sich in jedem Attribut einfügen.

InfoPath lässt die Ausführung von sogenannten „Managed Code“ zu. In InfoPath können die Sprachen C#, VB.NET sowie JScript<sup>15</sup> verwendet werden. Die Verwendung von benutzerdefinierten Code in einem InfoPath Formular kann unter anderem als Alternative zu den Regeln innerhalb eines Formulars genutzt werden. In einem früheren Abschnitt wurde beschrieben, wie Feldvalidierungen und die Anzeige von Hinweistexten mit Hilfe

---

<sup>15</sup> JScript ist eine objektorientierte moderne Skriptsprache, welche von Microsoft entwickelt wurde. Sie ist speziell für Anwendungen gedacht, die das .Net Framework nutzen und auf einem Server ausgeführt werden.

von Regeln umgesetzt wurden. Dasselbe Ergebnis ließe sich auch mit entsprechendem Programmcode ermöglichen. Managed Code kann im InfoPath über einen Code Editor dem Formular hinzugefügt werden. Als Code Editor dient Microsoft Visual Studio. Der Programmcode wird als \*.dll Datei in der Formularvorlagendatei gespeichert. Für die Bereitstellung von Formularen mit Programmcode ist ein SharePoint Administrator notwendig. Der SharePoint Administrator muss diese Formulare in eine Formularbibliothek von SharePoint hochladen und freigeben, bevor diese verwendet werden können <sup>16</sup>.

### 3.5 Anforderungen im E-Government

Allgemeine Anforderungen für Formulare im E-Government lassen sich nicht in einzelnen Punkten zusammenfassen. Oftmals spielen viele Faktoren eine Rolle, wodurch die Anforderungen von Fall zu Fall unterschiedlich sein können.

Ein wichtiger Aspekt ergibt sich aus möglichen Anforderungen aus rechtlicher Sicht. Aus den Webformularen werden elektronische Dokumente erstellt. Oftmals ist es notwendig, den Ausfüllenden des Formulars zu identifizieren und die Integrität des Dokuments sicherzustellen. Um dies zu überprüfen werden elektronische Signaturen verwendet. Die elektronische Signatur erfüllt den Zweck einer Unterschrift in Online-Behördengängen. Im E-Government spielt dies eine wichtige Rolle, da die Dokumente im Zweifel als Beweismittel vor Gericht verwendet werden müssen. Das ist nur dann möglich, wenn diese elektronisch signiert sind. Das bedeutet in manchen Fällen ist es notwendig, die ausgefüllten elektronischen Formulare zu signieren.

Cit bietet dazu weitere Integrationskomponenten an. In InfoPath lassen sich ebenfalls Signaturen verwenden. InfoPath ermöglicht das Signieren von einzelnen Abschnitten oder des ganzen Formulars mit Hilfe einer digitalen Signatur. Digitale Signaturen beschreiben technische kryptografische Verfahren zum Signieren von Dokumenten. Die elektronische Signatur ist ein rechtlicher Begriff. Dazu müssen in den Formularoptionen entsprechende Einstellungen getroffen werden, welche Bereiche signiert werden sollen. Anschließend lässt sich ein Abschnitt ins Formular einfügen und in dessen Eigenschaften die Signatur aktivieren. Abbildung 48 zeigt eine Signatur im InfoPath Formular.

---

<sup>16</sup> Bei Verwendung von SharePoint 365 reicht es aus, das Formular auf normalen Weg zu publizieren (das Publizieren wird in Abschnitt 4.1.2 auf Seite 63 näher beschrieben)

### 3 Vergleich des modellbasierten Ansatzes mit InfoPath

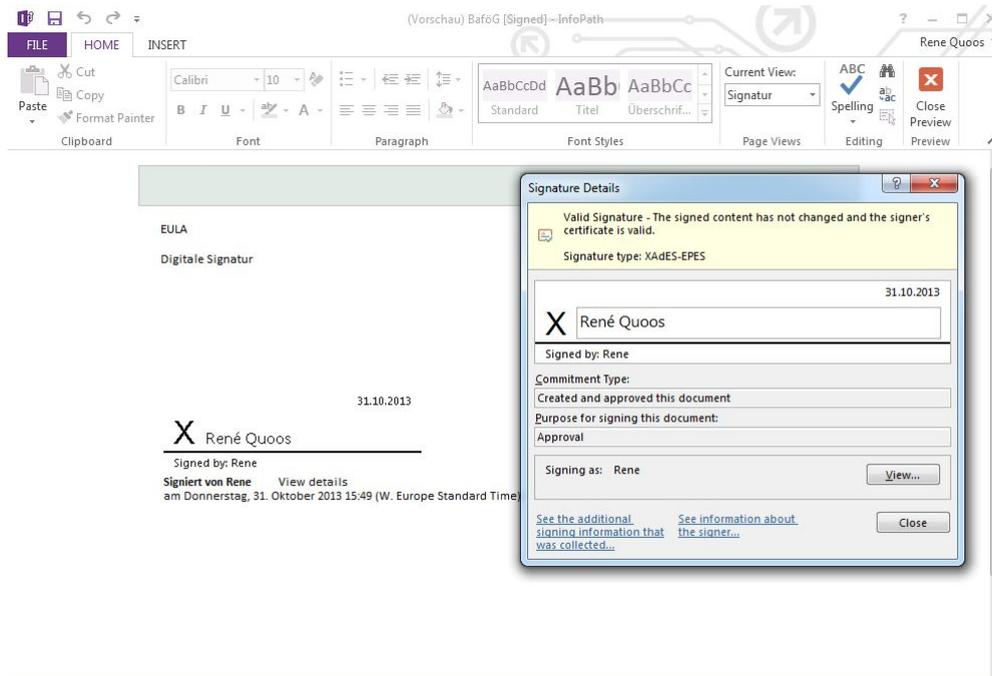


Abbildung 48: InfoPath - Digitale Signatur

Mit Hilfe der digitalen Signatur lässt sich die Identität des Antragstellers und die Integrität der Daten des Formulars prüfen. Ein anderer rechtlicher Aspekt ist die revisionssichere Speicherung der Formulare. Es gibt gesetzlich vorgeschriebene Aufbewahrungsfristen, die eingehalten werden müssen. Zur Aufbewahrung von elektronischen Dokumenten gibt es viele mögliche Ansätze. Es ist immer im Einzelfall zu betrachten, welche Anforderungen notwendig sind. Microsoft SharePoint bietet Möglichkeiten zur Verwaltung von elektronischen Dokumenten.

Eine weitere Anforderung in der Verwendung von Webformularen ist die Browserunabhängigkeit. Mit Browserunabhängigkeit ist gemeint, dass die Formulare sich nicht nur in einem speziellen Webbrowser anzeigen lassen, sondern unabhängig vom Browser korrekt dargestellt werden. Die Formulare werden auf einem Java basierten Anwendungsserver ausgeführt. Der notwendige Javascript Code wird serverseitig ausgeführt. Dies hat den Vorteil, dass die Formulare in allen gängigen Browsern korrekt dargestellt werden. Browserfähige InfoPath Formulare werden mit Hilfe der Forms Services auf einem SharePoint Server ausgeführt. Sie lassen sich ebenfalls in allen gängigen Webbrowsern ausfüllen. Einige Einschränkungen gibt es jedoch bei browserfähigen InfoPath Formularen. Einerseits können keine Hilfetexte in separaten Browserfenster angezeigt werden. Zu den Formularen lassen sich ganze Hilfs-Assistenten, die eine ausführliche Erklärung zu Inhalten des Formulars enthalten (z. B. gesetzliche Vorschriften), erstellen. Diese

werden beim Klick auf ein Info-Symbol im Formular in einem kleinen Pop-Up-Fenster geöffnet. In InfoPath ist das Öffnen solcher Dialogfenster nicht möglich. Andererseits gibt es Einschränkungen bei der Verwendung von digitalen Signaturen. Die Steuerelemente für das digitale Signieren sind sogenannte ActiveX-Steuerelemente. ActiveX ist eine von Microsoft entwickelte Softwarekomponente für aktive Inhalte. Diese Steuerelemente funktionieren nur im Microsoft Internet Explorer. In anderen Browsern funktionieren diese Steuerelemente nicht bzw. nur mit Hilfe von zusätzlichen Browsererweiterungen.

#### **3.6 Vorteile des modellbasierten Ansatzes**

In den vorherigen Abschnitten wurden ausgewählte Aspekte zweier Werkzeuge zur Formularerstellung miteinander verglichen. Dies geschah zunächst aus einer objektiven Sichtweise ohne Wertung. In diesem Abschnitt geht es darum, den Vergleich zu bewerten. Es sollen die Vorteile einer modellbasierten Generierung von InfoPath Formularen aufgezeigt werden.

Ein wichtiges Kriterium für die Bewertung ist die Art und Weise der Erstellung der Formulare. Im cit Modell geschieht eine strikte Trennung von Inhalt und Design des Formulars. Der Modellierer bildet das Formular in einer abstrakten Form ab und beschränkt sich lediglich auf inhaltliche und logische Zusammenhänge des Formulars. Das eigentliche Aussehen des Formulars wird vom IntelliForm Server zentral mit Hilfe von Stylesheets verwaltet. In InfoPath hingegen wird das Formular auf dem Zeichenblatt so erstellt, wie es später auch aussehen soll. Die gesammelten Erfahrungen bei der Erstellung des BAföG-Formulars in InfoPath haben gezeigt, dass beim Erstellen sehr viel Zeit für Designanpassungen notwendig ist. Das lenkt den Formularersteller vom eigentlichen Inhalt des Formulars ab und macht eine Formularerstellung zeitaufwendig. Sicherlich können durch Verwendung von Vorlagen und definierte Designs einige Anpassungen am Aussehen schneller vorgenommen werden, jedoch bleiben manuelle Anpassungen am Aussehen nicht aus. Ein Beispiel für die Notwendigkeit dieser Anpassungen ist das gewählte Tabellendesign im Zusammenhang mit dem Ein- und Ausblenden von Steuerelementen. Damit der Formular-Ausfüller später nicht merkt, dass sich die Ansicht aus mehreren verschiedenen Bereichen zusammensetzt, müssen die einzelnen Tabellen innerhalb dieser Bereiche angepasst werden. Das bedeutet, dass Anpassungen an Seitenabstände, Spaltenbreiten, Zeilenhöhen und Hintergrundfarben vorgenommen werden müssen. Ebenfalls muss die Größe der Eingabefelder angepasst werden. Standardmäßig wird die Breite ei-

nes Steuerelements auf 100% festgelegt. Im Tabellendesign sollte daher entweder die Breite der Spalte mit den Steuerelementen oder die Breite der Steuerelemente angepasst werden. Ein Vorteil der WYSIWYG-Erstellung ist, dass der Formular-Ersteller das Aussehen schneller und ohne Programmierkenntnisse anpassen kann und die Änderungen direkt sichtbar sind. Gleichzeitig ermöglicht diese Freiheit aber auch das Abweichen von einem einheitlichen Design aller Formulare.

Gemeinsamkeiten sind in den Eigenschaften der Steuerelemente zu finden. In cit Modelle werden Eigenschaften und Verhaltensweisen der Eingabefelder und Sätze über Attribute gesteuert. In InfoPath ist dies im Prinzip nichts anderes. Jedoch bietet InfoPath eine benutzerfreundlichere Oberfläche und Assistenten, die ihn dabei unterstützen. Wichtige Eigenschaften von Steuerelementen, wie zum Beispiel der Datentyp oder die Markierung als Pflichtfeld, lassen sich zum Beispiel einfach über die Ribbon ändern. Eine Erstellung der Formulare ohne Kenntnisse zu weiteren Technologien oder Programmierkenntnisse ist möglich. Mit zunehmender Komplexität des Formulars, speziell bei Verwendung von Regeln, um zum Beispiel eine Altersvalidierung vorzunehmen oder im Beispiel aus Abschnitt 3.1.3 auf Seite 35 zur Erstellung einer Verteilerseite bei Verwendung von Unterformularen, werden diese jedoch erforderlich. In cit Modell erfordern viele Attribute als Wert einen Javascript Ausdruck. Dies macht die Modellierung eines Formulars als Ganzes mit cit ohne Programmierkenntnisse sehr schwer bis hin zu unmöglich. An dieser Stelle muss allerdings auch die Zielgruppe, die mit den Composer angesprochen wird beachtet werden. In der Regel werden cit Formulare durch geschulte Mitarbeiter erstellt. Diese sind für die fachliche Modellierung zuständig, soweit ihre Kenntnisse ausreichen. Die Modellierung der komplexeren Logik des Formulars wird hauptsächlich durch Partnerunternehmen von cit vorgenommen, die entsprechende Kenntnisse besitzen, um die Logik mit Java abzubilden. Mit zunehmender Komplexität sind auch in InfoPath weiterführende Kenntnisse erforderlich.

Ein weiteres Kriterium in der Bewertung ist die Übersichtlichkeit. Je größer ein Formular wird, desto schwieriger ist es die Übersicht zu behalten. In einem Modell werden die einzelnen Dialogseiten abstrahiert auf einem Zeichenblatt dargestellt. Die Zusammenhänge zwischen einzelnen Dialogseiten und die Zuordnung von Formularfeldern zu den Dialogseiten wird über Pfeile dargestellt. Durch diese Art der Darstellung lassen sich im Vergleich zu InfoPath die Zusammenhänge besser erkennen. In InfoPath werden Zusammenhänge zwischen den einzelnen Ansichten nicht grafisch dargestellt. Lediglich eine

Auflistung aller vorhandenen Ansichten ist gegeben (siehe Abb. 35 auf Seite 38). Bei kleineren Formularen mit wenigen Ansichten mag dies weniger problematisch sein, jedoch geht bei großen Formularen wie dem BAföG-Antrag der Zusammenhang zwischen den einzelnen Seiten verloren. Das im Vergleich erstellte InfoPath-Formular stellt lediglich einen Teil des Antrages dar. Das heißt ein vollständiges BAföG-Formular würde weitaus mehr Ansichten beinhalten.

Feldvalidierungen, Regeln und Bedingungen sind ein wesentlicher Bestandteil von interaktiven Formularen. Durch sie erhalten Formulare ihre Dynamik, indem der Benutzer auf Fehleingaben hingewiesen wird oder der Dialogverlauf in Abhängigkeit von ausgefüllten Formularfeldern beeinflusst wird. Eine typspezifische Feldvalidierung wird sowohl im cit als auch im InfoPath Formular über den Datentyp vorgenommen. Alle zusätzlichen Feldvalidierungen werden im cit Formular anhand gesetzter Attribute vorgenommen. Die Werte dieser Attribute werden meist durch Javascript ausgedrückt. Schlägt eine Feldvalidierung im Ausfüllprozess fehl, werden automatisch entsprechend definierte Fehlermeldungen im Formular deutlich angezeigt. In InfoPath muss eine Zusammenfassung der Validierung (Validation Summary) entweder mit einem Konstrukt von mehreren Regeln an verschiedenen Steuerelementen oder mit Programmcode erzeugt werden. Aus Sicht des Formular-Erstellers ist die vordefinierte Lösung wesentlich einfacher, da der Modellierer sich zum Beispiel keine Gedanken über Schaltflächen für den Dialogverlauf machen muss. Im InfoPath müssen die Ansichtenwechsel per Regeln an Schaltflächen erstellt werden. Soll verhindert werden, dass der Benutzer zur nächsten Ansicht wechseln kann, bevor alle Eingaben korrekt ausgeführt wurden, muss dies ebenfalls per Regeln abgefangen werden. Das bedeutet der Formular-Ersteller muss sich genaue Gedanken über die Verwendung von Regeln machen. Dies kann unter Umständen sehr komplex werden. Eine Generierung aus einem Modell hätte den Vorteil, dass die Schaltflächen zum Ansichtenwechsel und dessen Logik anhand von gesetzten Attributen automatisch erstellt werden. Dies erspart nicht nur Zeit und reduziert Fehler, sondern ermöglicht den Formular-Ersteller sich auf die wesentlichen Inhalte des Formulars zu konzentrieren.

Einen wesentlichen Teil zur Übersichtlichkeit trägt die Modularisierung bei. Durch die Modularisierung lassen sich häufig verwendete Bausteine erstellen und in anderen Formularen wiederverwenden. Diese Bausteine können an einem zentralen Ort gespeichert und verwaltet werden. Änderungen der Bausteine lassen sich durch eine zentrale Verwaltung einfacher aktualisieren. In InfoPath sind lediglich einfache Bausteine zur Wiederver-

wendung erstellbar. Modelle bieten eine weitaus tiefere Modularisierung der Formulare. Soll zum Beispiel ein Satz in einem Formular mehrfach verwendet werden, lässt sich dies durch Setzen eines einzelnen Attributes an den jeweiligen Relationen zum betreffenden Satz realisieren. In InfoPath würde dies eine mehrfache Erstellung der Felder mit unterschiedlichen Bezeichnungen (Bezeichnungen der Steuerelemente müssen eindeutig sein) bedeuten. Ein anderes Beispiel ist die Verwendung von Verfeinerungen. Diese erhöhen die Übersichtlichkeit insofern, dass Details auf ein anderes Zeichenblatt oder gar in eine andere Datei verlagert werden. Somit wird das Formular auf den führenden Zeichenblatt kompakt und allgemein gehalten, während sich die Details auf anderen Zeichenblättern befinden. Die Verfeinerungen selbst können wiederum auch sehr komplex sein und weitere Verfeinerungen enthalten. Das Aufteilen von großen Formularen in mehreren Dateien verbessert zudem die Möglichkeit der verteilten Modellierung, d.h. mehrere Personen können gleichzeitig an einem Formular arbeiten. Dies ist zwar in InfoPath mit den Template Parts auch möglich, jedoch, wie bereits erwähnt, nur für einzelne Ansichten.

Formularmodelle könnten, ähnlich wie Prozessmodelle der Prozessmodellierung, in einem Portal zentral verwaltet werden. Durch SemTalk wird die Erstellung eines solchen Portals (z. B. in SharePoint) unterstützt. Eine Formularlandkarte könnte Aufschluss darüber geben, welche Formularbausteine (externe Verfeinerungen) in welchen Formularen verwendet worden sind. Eine solche Übersicht ist besonders dann wichtig, wenn Bausteine aktualisiert wurden und dementsprechend auch in den betreffenden Formularen Aktualisierungen vorzunehmen sind.

Externe Daten und Schnittstellen zu anderen Systemen spielen eine nicht zu unterschätzende Rolle für Formulare. Die Formularerstellung ist ein Aspekt im gesamten Formularmanagement. Zur Formularerstellung gehören auch Möglichkeiten zum Verwenden von externen Daten, zum Beispiel aus vorhandenen XML-Dateien, Webservices, Datenbanken oder anderen Systemen. Im cit Modell werden solche externe Ressourcen und Verbindungen zu Webservices durch Shapes dargestellt. Die Verbindungen zu diesen Daten werden mit der Angabe von URLs und XPath-Ausdrücken in den entsprechenden Attributen hergestellt. InfoPath bietet zum Einbinden solcher Ressourcen eine benutzerfreundlichere Oberfläche in Form von Assistenten, die den Benutzer Schritt für Schritt bei der Integration externer Daten oder Schnittstellen zu anderen System wie SharePoint unterstützen.

Zusammenfassend lässt sich sagen, dass ein modellbasierter Ansatz zur Generierung von InfoPath Formularen diverse Vorteile in der Erstellung, Übersichtlichkeit, Modularisierung und in der Anwendung der Formularlogik aufweist. Die wesentlichen Unterschiede zwischen dem Modell und InfoPath liegen vor allem in der Darstellung des Formulars als Ganzes. Im Modell wird das Formular abstrakt dargestellt und der Inhalt strikt vom Aussehen getrennt. Logiken wie Ansichtenwechsel und Feldvalidierungen werden anhand von Pfeilen und Attributen an den Relationen und Shapes generiert. Im InfoPath müssen diese Zusammenhänge und Logiken vom Formularersteller selbst angelegt werden. Das erfordert vor allem Zeit und senkt die Übersichtlichkeit.

## 4 Anwendungsszenario mit SharePoint 2013

In den bisherigen Abschnitten wurde lediglich auf die Erstellung der Formulare eingegangen. Die Erstellung der Formulare ist jedoch nur ein Bestandteil eines gesamten Prozesses. Dieser Abschnitt soll das Gesamtbild verdeutlichen.

Formulare werden zum Erfassen von Informationen verwendet. Das Erfassen von Informationen allein, stellt den Anfang eines Prozesses dar. Mit dem Einreichen eines ausgefüllten Formulars wird ein Bearbeitungsprozess angestoßen. Die Bearbeitung des Formulars kann mehrere Stufen oder Instanzen durchlaufen, bis diese abgeschlossen ist. Webformulare haben den Vorteil, dass die Informationen in elektronischer Form eingereicht werden können. Dies vereinfacht eine elektronische Bearbeitung der Formulare durch automatisierte Mechanismen.

Der IntelliForm Composer für Formular-Assistenten stellt nur die Komponente zur Generierung von Formular-Assistenten der IntelliForm Produktfamilie dar. Die IntelliForm Produktfamilie beschreibt eine ganzheitliche Lösung vom Erstellen bis hin zur Verarbeitung elektronischer Formulare. Die Vorteile elektronischer Formulare liegen nicht nur auf der Seite des Antragsstellers sondern auch auf der Seite der Antragsbearbeitung. Ein angeschlossenes Antrags- u. Fallmanagement-System bietet eine computergestützte Verarbeitung elektronisch eingereicherter Formulare. Dieser Abschnitt beschreibt die Workflow-Management-Komponente aus Abbildung 11 auf Seite 19 näher.

Workflows dienen der strukturierten Steuerung von Vorgängen. Der Workflow-Prozess beschreibt alle Aktivitäten eines Vorgangs ([ITW13c]). Mit IntelliForm Spaces werden die Funktionalitäten des IntelliForm Servers erweitert, um das Umsetzen von formularzentrischen Prozessen zu gewährleisten. Diese Prozesse lassen sich mit dem IntelliForm Generator erstellen (Abb. 49). Ähnlich wie bei den Formularen werden die Verwaltungsprozesse in einem SemTalk Modell beschrieben und aus ihnen die Logik zur Ausführung generiert. Das Modell enthält eine semantisch-deklarative Beschreibung des Prozesses. Im Modell enthalten sind Informationen wie beteiligte Verwaltungsdienstleistungen mit Ein-/Ausgangsdokumente, Formulare oder Metadaten.

Das Fallmanagement-System steuert den gesamten Genehmigungsprozess vom Ausfüllen bis hin zur Verarbeitung der Formulare. Im Fallmanagementsystem von Cit wird mit dem Einreichen ein Fall erstellt. Ein Fall löst einen entsprechenden Bearbeitungspro-

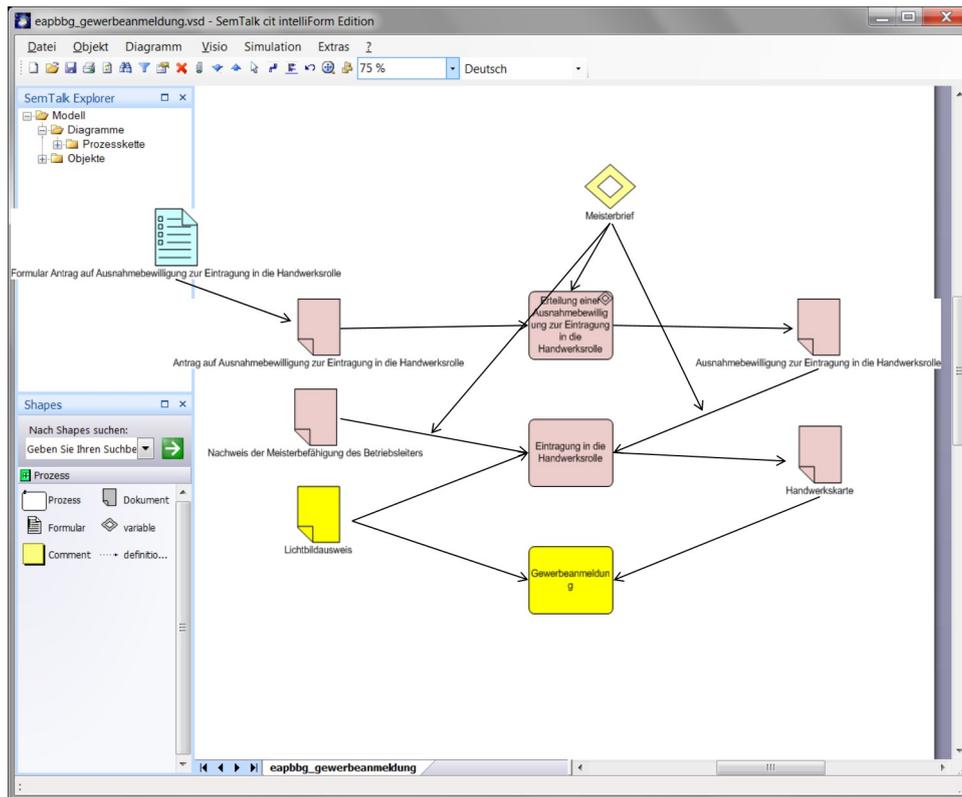


Abbildung 49: cit intelliForm Generator [cit13a]

prozess aus, der durch das System unterstützt wird. Im Wesentlichen steuert das Fallmanagementsystem die Kommunikation zwischen allen Beteiligten im Prozess. Abbildung 50 zeigt ein praktisches Beispiel aus dem Fallmanagementsystem. Zu jedem Fall werden fallspezifische Informationen sowie Checklisten für weitere Dokumente, die zur Verarbeitung eines Antrages benötigt werden, angezeigt. Es steuert ebenfalls den Ablauf des Genehmigungsprozesses. Dazu gehört unter anderem, welche Stellen im Genehmigungsprozess beteiligt sind und wer auf welche Daten zugreifen kann.

Cit intelliForm ist ein sehr komplexes System für das Antrags- bzw. Fallmanagement. Die einzelnen Prozesse lassen sich aus einem Modell heraus generieren und auf dem intelliForm Server entsprechend ausführen. Es gilt, diese Fallbearbeitung auf die Microsoft/SharePoint Welt zu übertragen. Das bedeutet, dass die Formulare und die Prozesssteuerung in SharePoint abgebildet werden müssen. Nachfolgend soll ein Anwendungsszenario unter Verwendung von Microsoft Technologien skizziert werden. Das Anwendungsszenario beinhaltet die Integration eines InfoPath Formulars in SharePoint und eine Antragsbearbeitung mit Hilfe eines SharePoint Workflows. Anschließend soll thematisiert werden, welche Möglichkeiten bestehen, vorhandene SharePoint-Strukturen bzw. Formular-/Prozessmodelle zu nutzen. Dies beinhaltet zum einen die Fragestellung, wel-

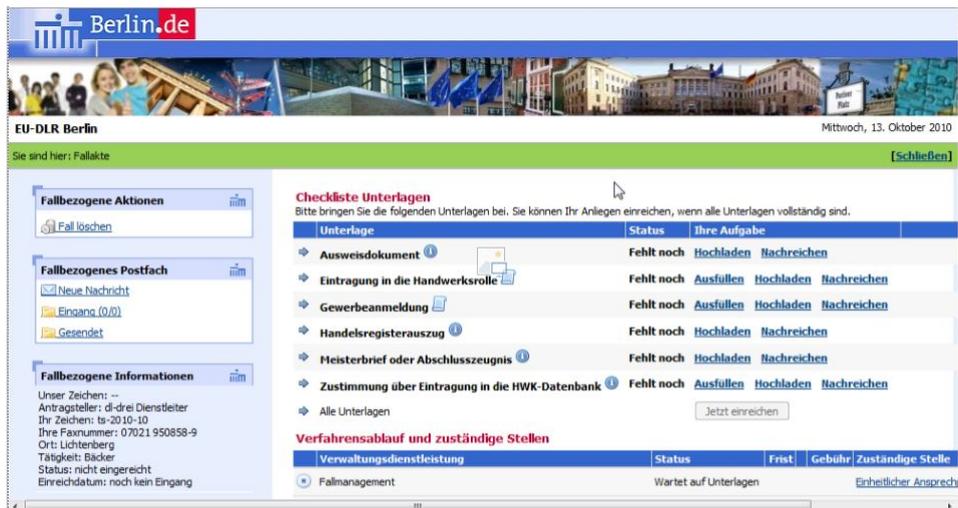


Abbildung 50: Fallmanagementsystem - Beispiel EU-DLR Berlin [cit13a]

che Informationen lassen sich aus einer vorhandenen SharePoint-Struktur ableiten, um daraus ein Formulardesign zu generieren und zum anderen, wie können vorhandene Modelle genutzt werden um eine entsprechende SharePoint Seitenstruktur bzw. Workflows zu erzeugen.

### 4.1 Beispiel: Brillenshop

Das Anwendungsszenario soll einen vereinfachten Brillenshop darstellen. Er soll den Kunden das Beantragen einer neuen Brille ermöglichen. Mit der elektronischen Einreichung des Formulars soll ein Bearbeitungsprozess gestartet werden. Der Bearbeitungsprozess stellt einen einfachen Genehmigungsprozess über zwei Stufen dar.

Für das Anwendungsszenario wurde vor dem Erstellen des Formulars eine Webseite im SharePoint angelegt, die die Oberfläche des Brillenshops bereitstellen soll (Abb. 51). Die Webseite enthält Listen zur zentralen Pflege von Krankenkassen, Versicherungsarten, Beschädigungsgründe sowie Hersteller, die als Datenquelle für die Auswahlfelder des Formulars dienen. Eine Aufgabenliste verwaltet Aufgaben, die den Mitarbeitern des Brillenshops zugeordnet werden können. Sie wird später im Workflow wichtig. Weiterhin wurden zwei Webpart-Seiten (*Kunde (Formular)* und *Sachbearbeiter*) im SharePoint angelegt. Die Webpart-Seiten sollen später eine Kundensicht bzw. eine Sicht des Sachbearbeiters repräsentieren.

Zusätzlich zu den Listen und Webpart-Seiten wurden zwei Bibliotheken angelegt. Die

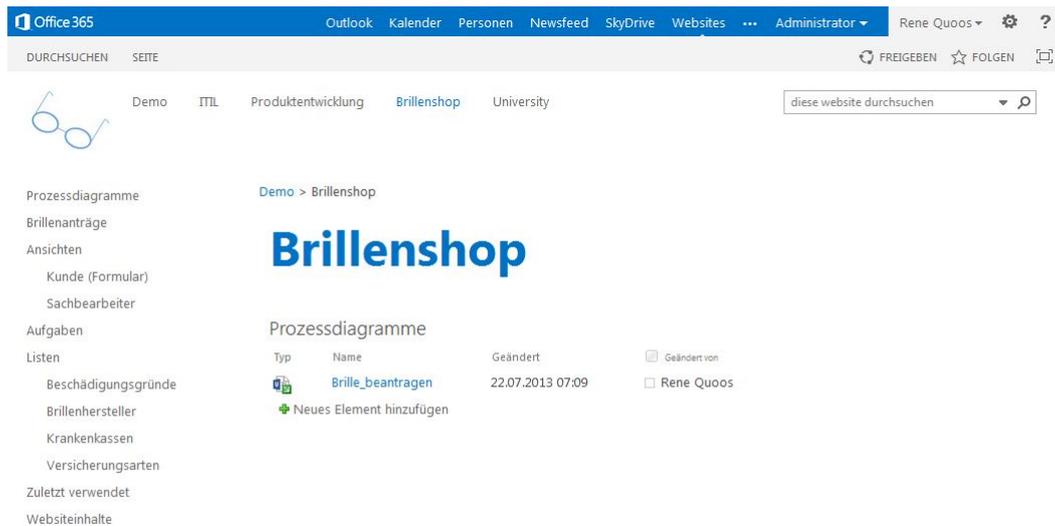


Abbildung 51: Seitenstruktur in SharePoint

Bibliothek „Prozessdiagramme“ dient lediglich als Ablage des cit Formulardesigns zum Brillenantrag. Die Bibliothek „Brillenanträge“ ist viel wichtiger. Diese Bibliothek ist eine spezielle *Formularbibliothek*, in der die Brillenantragsformulare veröffentlicht werden sollen.

### 4.1.1 Das Antragsformular

Das Formular zum Beantragen einer Brille wurde in Anlehnung eines bereits vorhandenen cit Modells, mit InfoPath erstellt. Es enthält mehrere Seiten zum Sammeln von Informationen bezüglich des Antragstellers und zur kaputten Brille. Die erste Seite des Formulars (Abb. 52) ist eine Begrüßungsseite, die den Ausfüllprozess des Formulars und damit den Gesamtprozess des Brillenantrages startet.

Auf den beiden folgenden Dialogseiten muss der Benutzer Angaben zu seiner Person (Abb. 53) und zur Brille (Abb. 54) machen.

Alle mit einem „\*“ gekennzeichneten Felder sind Pflichtfelder. Das Formular enthält mehrere Auswahlfelder (Krankenkasse, Versicherungsart, Beschädigungsgrund und Hersteller). Die Werte der Auswahlfelder werden über separate Share-Point-Listen gepflegt. Dies soll ein einfaches Hinzufügen von Auswahlmöglichkeiten über SharePoint, ohne dass das



Abbildung 52: Startseite des Formulars

Abbildung 53: Angaben zur Person

Formular selbst bearbeitet werden muss, ermöglichen. Mit Hilfe von Schaltflächen („Weiter“ und „Zurück“) gelangt der Antragsteller zur jeweils nächsten bzw. vorherigen Seite des Formulars.

Am Ende erhält der Antragsteller eine Übersicht seiner Eingaben (Abb. 55) und kann diese gegebenenfalls vor dem Absenden korrigieren.

Klickt der Antragsteller auf die „Absenden“-Schaltfläche wird eine Vorgangsnummer, welche sich aus dem aktuellem Datum und der aktuellen Zeit zusammensetzt, generiert und das Formular eingereicht. Der Antragsteller erhält eine Bestätigungsseite mit einer Vorgangsnummer (Abb. 56). Ein Klick auf die Schaltfläche „Beenden“ schließt das Formular.

**Brille beantragen**

**Angaben zur kaputten Brille**

Was ist passiert?  \*

Hersteller

Bitte machen Sie ein Foto Ihrer defekten Brille und laden Sie hier die Foto-Datei hoch.

Abbildung 54: Angaben zur Brille

**Zusammenfassung**

Bitte überprüfen Sie ihre Angaben vor dem Absenden auf Richtigkeit!  
Drucken Sie sich die Übersicht aus und legen Sie sie zu ihren Unterlagen.

**Angaben zur Person**

Geb Datum:  \*

Vorname:  \*

Nachname:  \*

Versicherungsnummer:  \*

Krankenkasse:  \*

Versicherungsart:  \*

**Angaben zur Brille**

Beschädigungsgrund:  \*

Hersteller:

Files:

Abbildung 55: Übersicht der Eingaben

Mit dem Absenden des Formulars wird selbiges in SharePoint unter Verwendung einer Kombination aus aktuellem Datum und aktueller Zeit als Dokumentnamen abgelegt.

### 4.1.2 Publizieren des Formulars auf Microsoft SharePoint 2013

InfoPath Formulare lassen sich auf einfachem Wege in SharePoint publizieren. Die Publizierung der Formulare wird vom InfoPath Designer durch einen Assistenten unterstützt. In den einzelnen Schritten des Assistenten werden einige wichtige Einstellungen vorgenommen. Dazu gehört neben der Option, dass das Formular im Browser ausgefüllt werden kann, auch die Auswahl, wie das Formular publiziert werden soll. Im Brillenshop-Beispiel

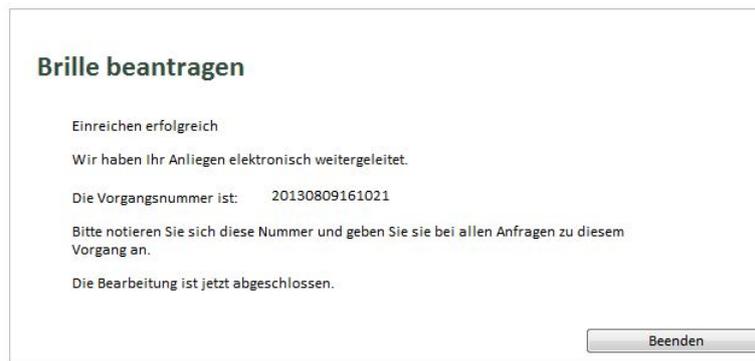


Abbildung 56: Bestätigungssseite

soll das Formular in einer Formularbibliothek, die zuvor im SharePoint angelegt wurde, publiziert werden. Damit das Formular über den Browser geöffnet und ausgefüllt werden kann, muss die entsprechende Option aktiviert werden. Eine weitere wichtige Einstellung ist die sogenannte „Property Promotion“. Die „Property Promotion“ legt fest, welche Eingabefelder des Formulars als SharePoint-Spalten in der Formularbibliothek angelegt werden.

Nach Abschluss des Publizierungsprozesses ist das Formular zur Verwendung im SharePoint bereit. Abbildung 57 zeigt die zuvor erstellte Formularbibliothek mit einigen Eingabefeldern des Formulars als SharePoint-Spalten. Wird der Bibliothek ein neues Dokument hinzugefügt, öffnet sich das Brillenantragsformular. Mit dem Absenden des Formulars wird ein neuer Eintrag in der Formularbibliothek erstellt.

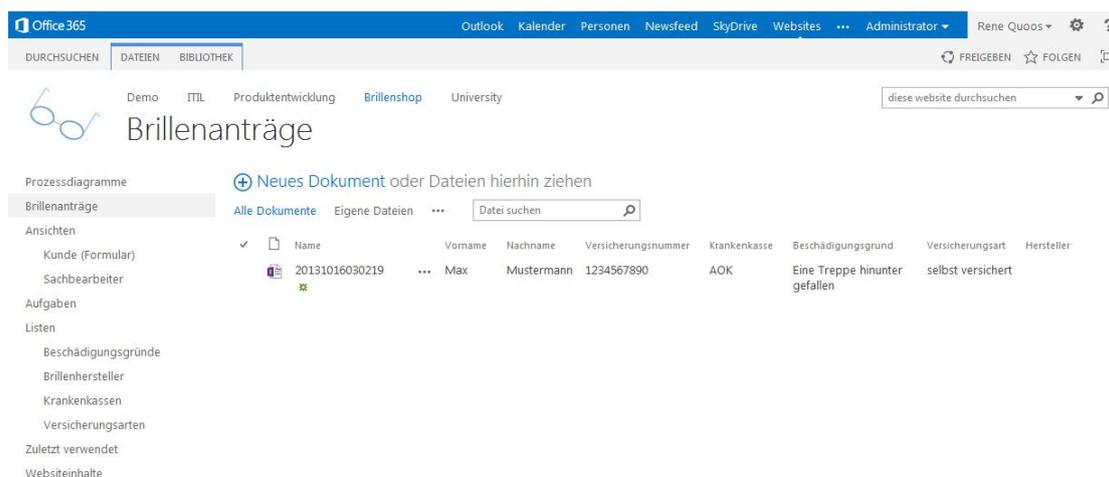


Abbildung 57: Publiziertes Formular in SharePoint

Eine Besonderheit von SharePoint ist die Bereitstellung von Inhalten über sogenannte Webparts. Mit Hilfe von Webparts lassen sich zum Beispiel Listen, Bibliotheken oder

auch InfoPath Formulare auf Webpart-Seiten einbinden. Dies bietet den Vorteil, dass sich die Formulare direkt in einer SharePoint Seite integrieren lassen und nicht über die Listenfunktion geöffnet werden müssen. Für den Brillenshop wurde eine entsprechende Webpart-Seite angelegt (Abb. 58). Diese Seite soll als Kundensicht dienen. Unter dem Formular werden eingereichte Anträge des Kunden in einer Listenansicht angezeigt. Die Liste dient vor allem zum Anzeigen des aktuellen Bearbeitungsstatus.

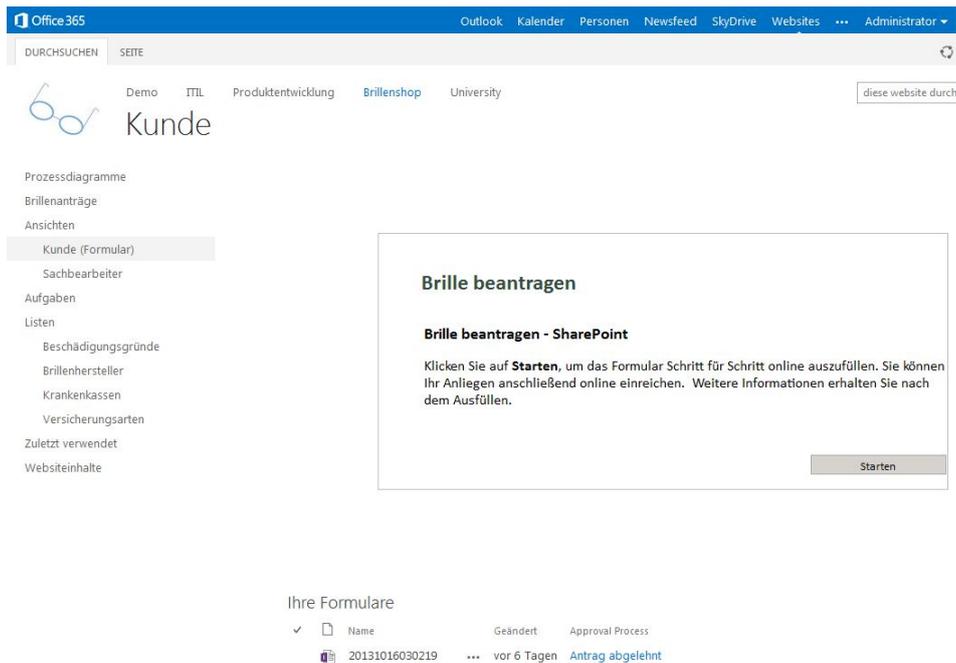


Abbildung 58: Brillenshop - Kundensicht

### 4.1.3 Verwendung von SharePoint Workflows

Das Einreichen eines Antragsformulares soll im Brillenshop einen Bearbeitungsprozess auslösen. Dieser Bearbeitungsprozess wird mit Hilfe eines SharePoint-Workflows realisiert. Im SharePoint-Umfeld bedeuten Workflows „...das automatisierte Verschieben von Dokumenten oder Elementen durch eine Abfolge von Aktionen oder Aufgaben, die zu einem Geschäftsprozess gehören.“ ([Mic13b]). SharePoint Workflows lassen sich mit dem SharePoint Designer erstellen. Der SharePoint Designer ist ein Tool, mit dem nicht nur SharePoint-Workflows erstellt, sondern auch ganze SharePoint Seiten angelegt und strukturiert werden können. Mit dem Tool können Workflows in einer textbasierten Ansicht (Abb. 59) oder in einer visuellen Ansichten modelliert werden.

Die Modellierung des Workflows in der visuellen Ansicht erfolgt mit einer im Share-

## 4 Anwendungsszenario mit SharePoint 2013

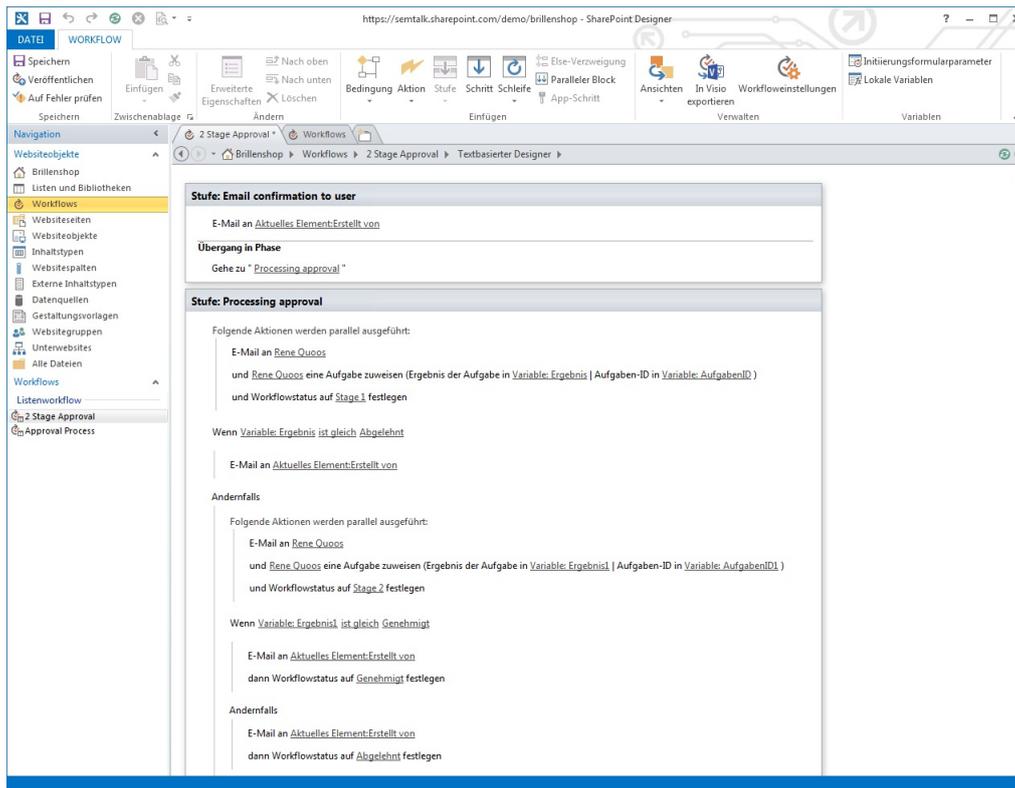


Abbildung 59: SharePoint Designer - Textbasierte Ansicht

Point Designer integrierten Visio-Oberfläche (ab SharePoint Designer 2013) und entsprechenden Shapes für den Workflow. Abbildung 60 zeigt das grafische Modell des Genehmigungs-Workflows für den Brillenantrag. Der Genehmigungsprozess wird mit dem Einreichen des Formulars gestartet. Als erstes erhält der Antragsteller eine E-Mail Benachrichtigung, dass der Antrag eingegangen ist und bearbeitet wird. Anschließend folgen die zwei Schritte des Genehmigungsprozesses. Im ersten Schritt werden zunächst mehrere Aktionen parallel ausgeführt. Ein Bearbeiter bekommt eine Aufgabe zur Prüfung des Antrages zugewiesen und erhält eine E-Mail Benachrichtigung zur Aufgabe. Außerdem wird der Workflowstatus, welcher in der Liste auf der Kundensicht angezeigt wird, auf einem bestimmten Wert festgelegt. Lehnt der Bearbeiter den Antrag ab, erhält der Antragsteller eine E-Mail Benachrichtigung, dass der Antrag abgelehnt wurde. Der Workflowstatus wird entsprechend auf „Abgelehnt“ gesetzt und der Workflow beendet. Falls der Antrag jedoch genehmigt wurde, erfolgt Schritt zwei des Genehmigungsprozesses. Der zweite Schritt ist analog zum ersten. Ein Bearbeiter bekommt eine Aufgabe zugewiesen, wird darüber benachrichtigt und der Workflowstatus wird aktualisiert. Nach Beendigung der Aufgabe (Antrag wird genehmigt/abgelehnt) erhält der Antragsteller eine entsprechende Benachrichtigung und der Beispiel-Workflow ist abgeschlossen.

## 4 Anwendungsszenario mit SharePoint 2013

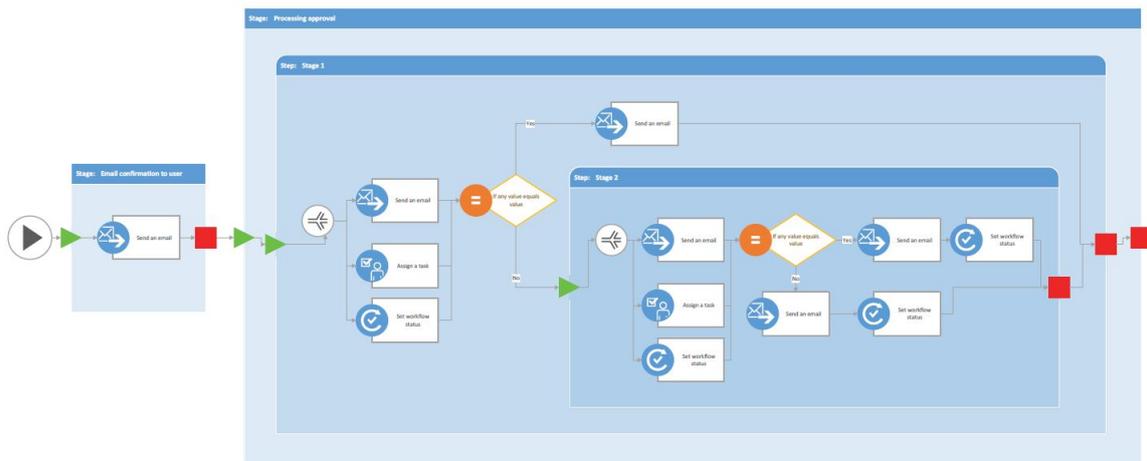


Abbildung 60: SharePoint Workflow - Visueller Designer

Zur Bearbeitung der Anträge dient die zuvor angelegte „Sachbearbeiter“-Seite (Abb. 61). Sie enthält als Webparts eingebunden die Aufgabenliste sowie die Liste mit vorhandenen Brillenanträgen. Die Aufgaben enthalten eine Verknüpfung zum entsprechenden Antragsformular, sodass der Bearbeiter diese aus der Aufgabe heraus öffnen kann. Die Liste der Brillenanträge könnte als Übersicht aller bearbeiteten Anträge bzw. noch zu bearbeitenden Anträge eines Sachbearbeiters dienen.

The screenshot shows the 'Sachbearbeiter' (Case Worker) page in SharePoint. The page title is 'Sachbearbeiter' and it includes a navigation menu with options like 'Prozessdiagramme', 'Brillenanträge', 'Ansichten', 'Kunde (Formular)', 'Sachbearbeiter', 'Aufgaben', 'Listen', 'Beschädigungsgründe', 'Brillenhersteller', 'Krankenkassen', 'Versicherungsarten', 'Zuletzt verwendet', and 'Websiteinhalte'.

The main content area is divided into two sections:

- Aufgaben (Tasks):** A table showing a list of tasks with columns for 'Vorgangname', 'Fälligkeitsdatum', 'Vorgangstatus', and 'Ergebnis der Aufgabe'. The tasks include 'Neuer Brillenantrag' (approved/rejected) and 'Approval Stage 2' (not started).
- Brillenanträge (Glasses Orders):** A table showing a list of glasses orders with columns for 'Name', 'Geändert', 'Geändert von', 'Vorname', 'Nachname', 'Versicherungsnummer', 'Krankenkasse', 'Beschädigungsgrund', and 'Versicherungsart'. The orders include details for '20131016030219' and '20131016072155'.

Abbildung 61: Brillenshop - Sachbearbeiter Sicht

### 4.2 Verwendung des SemTalk SiteBuilders

Das Brillenshop-Szenario wurde vollständig unter Verwendung von InfoPath, SharePoint und SharePoint Designer erstellt. Eine solche Vorgehensweise ist zeitaufwendig und wird mit zunehmender Komplexität noch zeitintensiver. Allein das Erstellen der SharePoint Listenstruktur mit entsprechenden Spalten nimmt eine gewisse Zeit in Anspruch. Eine genaue Planung, welche Listen benötigt und in welcher Reihenfolge diese angelegt werden, ist notwendig und erfordert viele Klicks in der SharePoint-Oberfläche. Für das Brillenshop Szenario wurde, anhand eines vorhandenen cit Formularmodells, eine entsprechende Lösung mit Microsoft Tools erstellt. Ein anderes Szenario wäre, wenn aus einer SharePoint Struktur, bestehend aus Listen und Spalten, ein Formularmodell abgeleitet werden soll. Die entscheidende Frage ist, wie sich vorhandene Datenmodelle nutzen lassen, um daraus Formularmodelle bzw. SharePoint Strukturen/Workflows zu generieren.

Eine mögliche Schnittstelle könnte der SemTalk SiteBuilder bereitstellen. Der SemTalk SiteBuilder ist ein Werkzeug von SemTalk, mit dessen Hilfe sich Seitenstrukturen in SharePoint aus einem Modell erzeugen lassen. Üblicherweise wird eine Seitenstruktur, d. h. Anlegen von Listen mit dazugehörigen Spalten, über die SharePoint Oberfläche aufgebaut. Der Benutzer legt sich seine gewünschte Struktur durch ein „Zusammenklicken“ der notwendigen Listen und dazugehörigen Spalten an. Bei größeren Seitenstrukturen erfordert dies viel Zeit und sehr viele Klicks an verschiedenen Stellen. Zudem muss der Benutzer eine bestimmte Reihenfolge beim Erstellen seiner Seitenstruktur einhalten. Der SiteBuilder ermöglicht es, mit Hilfe eines Modells, eine Seitenstruktur zu Beschreiben. Er stellt Shapes zur Verfügung, die die SharePoint Elemente repräsentieren. Listen lassen sich einfach erstellen und Spalten können mit Verbindungspfeilen den Listen zugeordnet werden. Abbildung 62 zeigt ein vereinfachtes SiteBuilder Modell für den Brillenshop. Es gibt eine Liste Brillenanträge, die die Daten des Brillenantrages hält. Einige Spalten dieser Liste sind als „Lookup“ modelliert und beziehen Auswahlwerte aus anderen Listen. Mit Hilfe des SiteBuilder Modells kann per Knopfdruck eine entsprechende Seitenstruktur mit den modellierten Listen und dazugehörigen Spalten erzeugt werden.

Die Erstellung einer solchen Seitenstruktur ist vor allem dann sinnvoll, wenn ausgefüllte cit Formulare in SharePoint integriert werden sollen (siehe Abschnitt 4.2.2 auf Seite 71). Bei Verwendung von InfoPath Formularen wie im zuvor beschriebenen Brillenshop-Beispiel aus Abschnitt 4.1 ist die Vorgehensweise etwas anders. Werden InfoPath Formu-

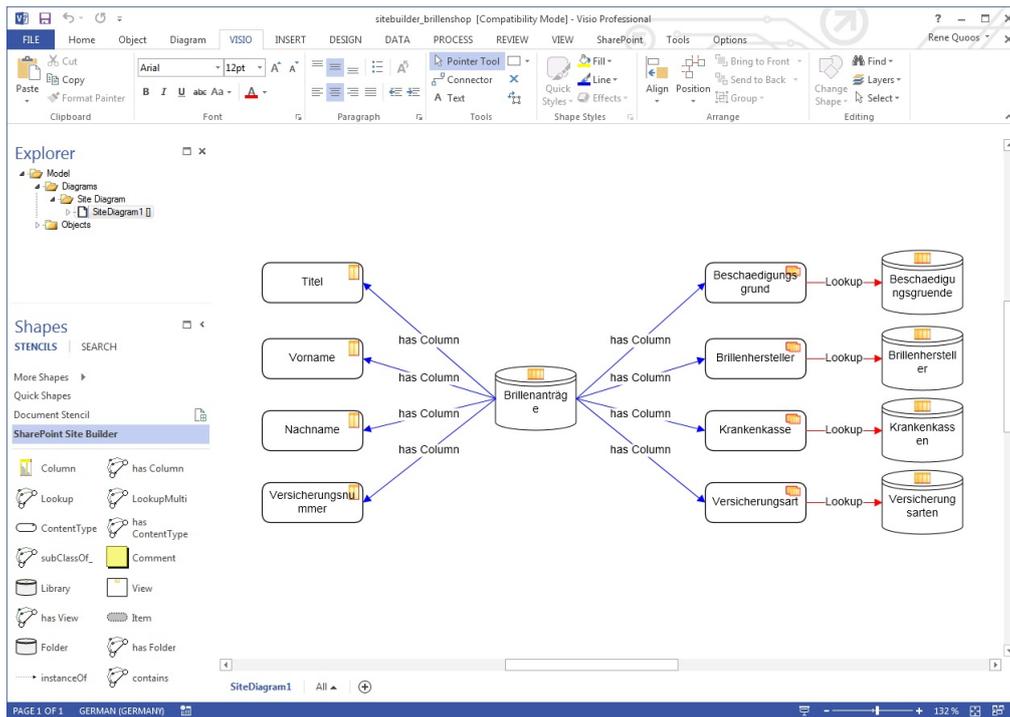


Abbildung 62: SemTalk Site Builder

lare verwendet, muss die Brillenantragsliste als Formularbibliothek angelegt werden. Der Formularbibliothek selbst, wird nur eine Spalte mit dem Titel des Antragsdokuments zugeordnet. Die restlichen Spalten (Vorname, Name, Versicherungsnummer und die Lookups auf andere Listen) werden vom InfoPath Formular mit der Property Promotion erzeugt und enthalten Formulardaten. Der Lookup für die Auswahlfelder für Krankenkasse, Versicherungsart, Beschädigungsgründe und Brillenhersteller erfolgt im Formular.

### 4.2.1 Ableitung eines Formulars aus einer SharePoint-Struktur

Betrachten wir das Brillenshop-Szenario aus einer anderen Sichtweise, nämlich dass die zuvor angelegte Seitenstruktur bereits im SharePoint existiert, das Formulardesign selbst jedoch noch nicht. Mit dem SiteBuilder lassen sich nicht nur Seitenstrukturen in SharePoint erstellen, sondern auch aus SharePoint in den SiteBuilder importieren. Die SharePoint Struktur wird in einem SemTalk Modell abgebildet. Dies könnte auch für die Erstellung der Formulare von Nutzen sein. Angenommen, es besteht bereits eine Seitenstruktur für den Brillenshop im SharePoint, der Brillenshop enthält eine Liste der Kunden mit den Spalten Name, Nachname, Krankenkasse, Versicherungsnummer und Versicherungsart. Für die Krankenkasse und die Versicherungsart existieren weitere Listen mit

vorhandenen Krankenkassen und Versicherungsarten. In SharePoint werden die beiden Spalten (Krankenkasse und Versicherungsart) als „Lookup“ angelegt. Ein Lookup verhält sich ähnlich wie ein Auswahlfeld. Die Lookup-Spalten beziehen ihre Werte aus den beiden Listen Krankenkassen und Versicherungsarten. Weiterhin existiert eine Liste mit Angaben zur Brille (wie Hersteller, Modellbezeichnung und Bild). Die Spalte Hersteller bezieht ihre Werte aus einer separaten Herstellerliste. Eine sechste Liste enthält Gründe, warum die Brille kaputt gegangen sein könnte.

Anhand dieser Seitenstruktur soll nun ein Formular erstellt werden. Ein Ansatz wäre die Erstellung in InfoPath, ein anderer die Verwendung des SiteBuilders. Mit dem SiteBuilder könnten die Listen und dessen Spalten als Modell importiert werden. Betrachten wir erneut das für den Brillenshop erstellte Formular aus Abschnitt 4.1. Es könnten folgende Rückschlüsse gezogen werden. Die Liste der Kunden ähnelt der zweiten Seite des Formulars. Die Liste mit den Angaben zur Brille ähnelt in etwa der dritten Seite des Formulars. Diese beiden Listen entsprächen in etwa einer Formularansicht. Die Lookup-Spalten könnten Auswahlfelder sein, dessen Werte in den anderen Listen stehen. In InfoPath entspricht das einer Datenverbindung, um eine Werteliste für ein Auswahlfeld aus SharePoint zu beziehen. In SharePoint lassen sich einzelne Spalten als Pflichtfeld markieren um Eingaben beim Anlegen eines neuen Listenelements zu erzwingen. SharePoint Spalten enthalten also auch Informationen, welche Eingabefelder mit einer Pflichtfeldüberprüfung zu versehen sind.

Weiterhin ermöglicht SharePoint die Verwendung von *Inhaltstypen* (engl. „Content Types“) für Listen. Inhaltstypen wären im Kontext des Brillenshops zum Beispiel verschiedene Arten von Brillen. Es gibt Lesebrillen, Gleitsichtbrillen, Sonnenbrillen usw. Während alle Brillen gemeinsame Attribute wie Modellbezeichnung oder Material besitzen, können für Gleitsichtbrillen zusätzliche Kennzahlen notwendig sein, die eine Lesebrille nicht hat. Die verschiedenen Brillenarten besitzen Gemeinsamkeiten und unter Umständen zusätzliche spezielle Eigenschaften. Das Brillenbeispiel ist ähnlich wie das Beispiel mit der Rechnungsanschrift und Lieferanschrift zu sehen. Eine Analogie zu den Bausteinvarianten des cit Composers ist durchaus zu erkennen. Dabei stellen die einzelnen Brillenarten die Varianten des Bausteins dar.

Der importierten SharePoint-Struktur fehlen allerdings noch die logischen Zusammenhänge zwischen den Listen, um daraus ein Formular mit miteinander verknüpften Ansichten zu machen. Im SiteBuilder Modell müssten demzufolge Relationen angelegt wer-

den, die die beiden Listen in eine logische Reihenfolge bringt. Zusätzlich könnte noch ein weiteres Element angelegt werden, um die Begrüßungsseite des Formulars abzubilden. Somit ließe sich im Prinzip aus einer SharePoint Struktur, bestehend aus Listen und Spalten, ein Formular ableiten. Neben den Informationen der einzelnen Ansichten und dessen zugeordneten Steuerelemente können Informationen zu Pflichtfeldern oder Datentypen aus SharePoint bezogen werden. Logische Abhängigkeiten zwischen Listen und zwischen den Eingabefeldern sind nachträglich zu modellieren.

Ein anderes Beispiel wäre, wenn als Ausgangssituation eine einzige Liste mit vielen Spalten existieren würde (zum Beispiel wie in Abschnitt 4.2 im SiteBuilder Modell des Brillenshops skizziert). Listenansichten in SharePoint ermöglichen das Anzeigen von ausgewählten Spalten einer Liste. In InfoPath werden Formularseiten als Ansichten bezeichnet. Das könnte bedeuten, dass anhand der erstellten Listenansichten sich ebenfalls Rückschlüsse auf eine Formularseite ziehen lassen. Die Betrachtungsweise aus dieser Sicht ist daher interessant, weil sich dadurch ein recht großer Teil des Formulars aus einer SharePoint Struktur ableiten ließe. Bereits vorhandene SemTalk Funktionalitäten könnten genutzt und erweitert werden.

Denkbar wäre auch eine Ableitung der SharePoint Struktur aus dem cit Formularmodell. Ähnlich, wie im vorherigen Szenario beschrieben, könnten einzelne Sätze als eigenständige Liste und Eingabefelder als Spalten angesehen werden. Bestimmte Eigenschaften zur Steuerung der Eingabefelder (z. B. Pflichtfeldmarkierung oder Reihenfolge) dienen als Vorlage für Einstellungen der einzelnen SharePoint Spalten. In diesem Szenario ließe sich ebenfalls der SemTalk SiteBuilder als Werkzeug zur Erstellung der SharePoint-Struktur nutzen. Anschließend könnte ein entsprechendes InfoPath-Formular generiert und in SharePoint integriert werden.

### **4.2.2 Integration der cit Formular-/Prozessmodelle in SharePoint**

Ein denkbare Szenario ist die Vermischung beider Technologien. Mit Vermischung ist gemeint, dass der eigentliche Dialogassistent weiterhin auf dem IntelliForm Server ausgeführt und als Webpart in SharePoint angezeigt wird. In diesem Szenario gilt es, die Java basierte JBoss Umgebung mit der von SharePoint zusammenzuführen. Eine Möglichkeit dies zu erreichen, ist die Verwendung vorhandener Schnittstellen. Sowohl SharePoint als auch IntelliForm stellen Schnittstellen bereit, um Daten auszutauschen. Der Austausch

der Daten wäre primär über Webservices zu realisieren. SharePoint stellt eine Reihe von Webservices bereit, die genutzt werden können um SharePoint Listeneinträge auszulesen bzw. zu erstellen. Der cit Composer bietet ebenfalls die Möglichkeit, über Schnittstellen Daten aus externen Quellen zu verwenden (z. B. Daten aus SharePoint Listen als Auswahlmöglichkeiten für Auswahlfelder) bzw. Daten eines ausgefüllten Formulars in einer SharePoint Liste zu speichern. Abbildung 63 zeigt eine schematische Darstellung, wie mit Hilfe von Webservices Daten zwischen intelliForm und SharePoint ausgetauscht werden könnten.

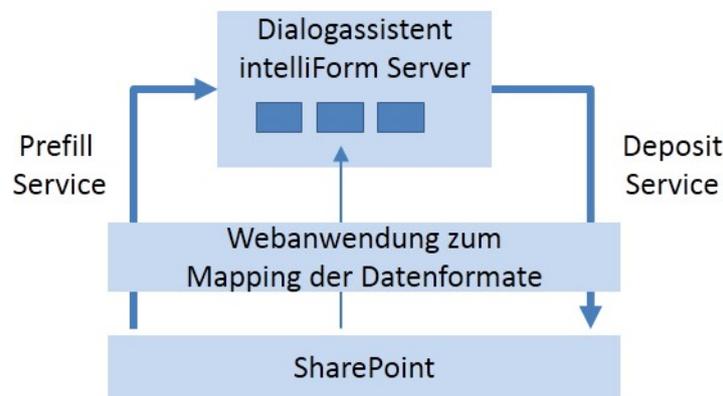


Abbildung 63: Datenaustausch zwischen intelliForm und SharePoint

IntelliForm stellt zwei Services bereit, die zum Datenaustausch genutzt werden können. Zum einen ist das der sogenannte „Prefill“ und zum anderen der „Deposit“. Eine Besonderheit der cit Dialogassistenten liegt in der Dynamik der Formulare. Mit Hilfe eines Prefills besteht die Möglichkeit, einen Dialogassistent mit Daten aus einer externen Datenquelle zu initialisieren. Das bedeutet, dass es möglich ist, Felder eines Dialog-Assistenten schon beim Initialisieren des Formulars zu füllen. Somit könnten zum Beispiel Auswahlmöglichkeiten für Auswahlfelder aus einer externen Quelle (zum Beispiel SharePoint) dynamisch herangezogen werden, anstatt die Auswahlmöglichkeiten fest im Dialogmodell zu verankern. Ein anderes Szenario im Zusammenhang mit SharePoint wäre zum Beispiel das Nutzen von Profilinformatoren, damit bestimmte Felder wie zum Beispiel Titel, Name, Vorname, Organisation oder sogar hinterlegte Adressen bereits im Vorfeld ausgefüllt sind. Der Deposit dient als Schnittstelle, um Daten des ausgefüllten Formulars in externe Quellen zu speichern. Im Brillenshop-Beispiel würden die Daten des ausgefüllten Formulars in einer Liste (Brillenanträge) abgelegt. Diese Möglichkeiten sind sehr vielversprechend, jedoch nicht ganz so trivial. Das Problem hierbei ist, die Daten entsprechend zu transformieren, damit SharePoint bzw. intelliForm die Daten nutzen kann.

Die Antragsbearbeitung könnte nach dem Speichern der Formulardaten in SharePoint mit entsprechenden Workflows erfolgen.

Mit Hilfe des intelliForm Generators lassen sich Workflows für das cit Fallmanagementsystem generieren. Ein beachtenswerter Aspekt ist, ob sich diese Prozessmodelle zur Modellierung von Workflows nutzen lassen, um daraus SharePoint Workflows zu generieren. Abbildung 64 skizziert zwei Vorgehensweisen zur Generierung der Workflows. Auf linken Seite ist die bisherige Variante zu sehen, in der aus den semantisch-deklarativen Prozessmodellen Java/JBoss basierte Workflows zur Ausführung auf dem intelliForm Server generiert werden. Die generierten Workflows lassen sich allerdings nicht in SharePoint verwenden. Ein Vorteil wäre, wenn das Prozessmodell zur Generierung der intelliForm Workflows genutzt werden könnten, um daraus SharePoint basierte Workflows zu generieren. SharePoint Workflows können auf verschiedene Art und Weise erstellt werden. Zum einen besteht die Möglichkeit, den Workflow mit dem SharePoint Designer zu modellieren und zum anderen ihn programmatisch zu erstellen. Eine weitere Möglichkeit, SharePoint Workflows zu beschreiben, bietet die Extensible Application Markup Language (XAML). XAML ist eine von Microsoft entwickelte XML-basierte Auszeichnungssprache zur Windows Oberflächengestaltung sowie zur Definition von Workflows. Mit Hilfe von XAML lassen sich Workflows deklarativ, d. h. ohne Programmcode, beschreiben und in Windows Workflow Foundation (WF)<sup>17</sup> ausführen. In Visual Studio bzw. in SharePoint Designer erstellte Workflows basieren letztendlich auch auf XAML. Der Vorteil von XAML ist, dass keinerlei Programmcode notwendig ist. Zur Generierung eines SharePoint Workflows muss lediglich ein XAML-konformes XML-Dokument erstellt werden. Im Prinzip müsste also „nur“ eine Export-Schnittstelle geschaffen werden, die ein XAML-Dokument erzeugt.

Die Integration der auf intelliForm Server ausgeführten Formulare in SharePoint mittels Webservices und die Generierung von Workflows aus den cit Prozessmodellen ist ein sehr komplexes Thema. Eine nähere Betrachtung der Integration von externen cit Formularen, wie dem Erstellen einer entsprechenden Webservice Schnittstelle sowie die Generierung von SharePoint-Workflows aus cit Modellen ist nicht mehr Teil dieser Arbeit. Diese Arbeit betrachtet hauptsächlich die Erstellung von Formularanwendungen mit Mi-

---

<sup>17</sup> Die Windows Workflow Foundation ist eine Workflow-Anwendungssoftware und ist Bestandteil von SharePoint ab Version 2007. SharePoint 2013 Workflows werden außerhalb von SharePoint ausgeführt (Workflow Manager 1.0 Server)

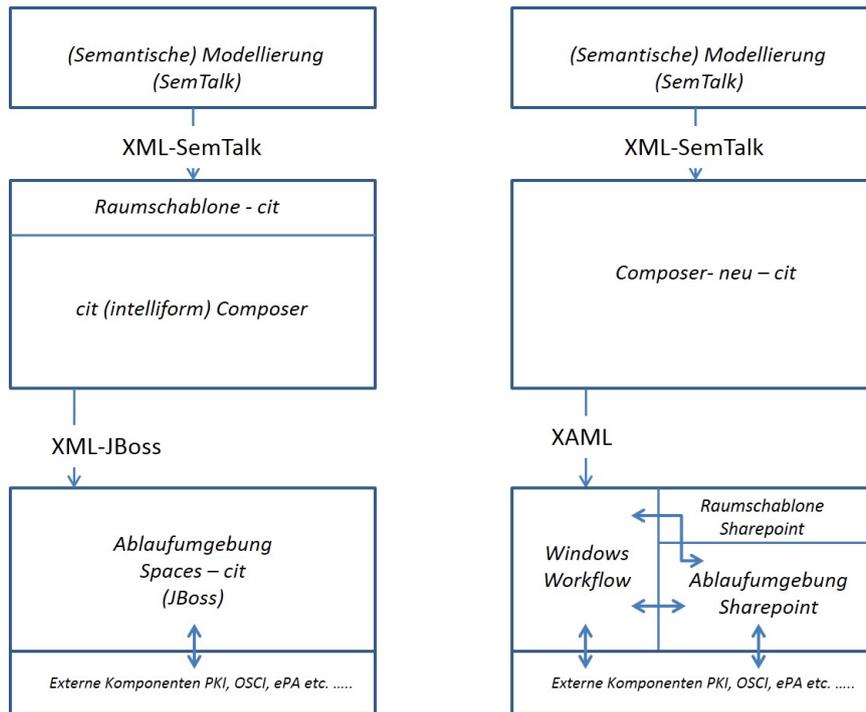


Abbildung 64: Modellbasierte Workflowgenerierung [cit13a]

Microsoft Tools. Es sollte lediglich skizziert werden, wie cit Formulare und Anwendungen mit SharePoint zusammengeführt werden könnten.

## 5 Generierung des Formulars aus dem Modell

In den vorherigen Abschnitten wurde eine konzeptionelle Betrachtung der Überführung eines cit Formulardatei nach InfoPath durchgeführt. Dieser Abschnitt soll einen Einblick auf die technische Sicht geben. Für die technische Umsetzung ist eine nähere Betrachtung der Struktur eines InfoPath Formulars notwendig. Im Folgenden wird etwas auf die XML-basierte Zusammensetzung des InfoPath Formulars eingegangen, um anschließend mit Hilfe eines Prototyps eine Anwendung zu entwickeln, welche aus dem Formulardatei ein natives InfoPath Formular erzeugt. Im Vordergrund steht hierbei nicht die Entwicklung des Prototyps, sondern vielmehr das Aufzeigen der Machbarkeit und die Einschätzung des Zeitaufwandes für eine mögliche Umsetzung.

### 5.1 Aufbau des InfoPath Formulars

Interessant für die technische Umsetzung ist, wie InfoPath ein Formular abspeichert. InfoPath speichert Formulare als Formularvorlagendatei ab. Die Formularvorlagendatei ist eine einzige Datei, welche mehrere Hilfsdateien enthält. Das Erscheinungsbild, die Datenstruktur sowie das Verhalten fertig gestellter Formulare wird durch die XSN-Datei definiert ([Mic13a]). Um sich den genauen Inhalt der Vorlagendatei anschauen zu können, muss die Dateinamenserweiterung von „.xsn“ in z. B. „.cab“<sup>18</sup> geändert werden. Anschließend lässt sich die Vorlagendatei im Windows Explorer öffnen. Die Vorlagendatei enthält mindestens die in Abbildung 65 abgebildeten Dateien.

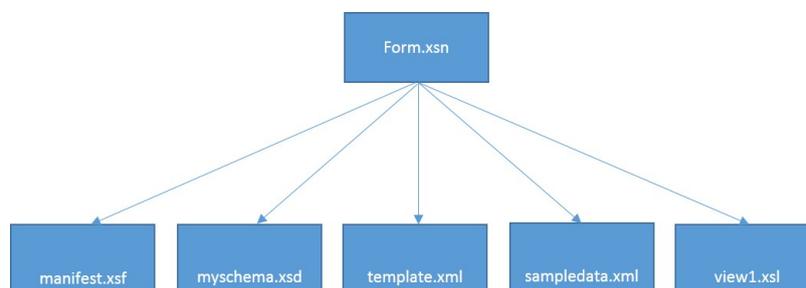


Abbildung 65: Zusammensetzung der InfoPath Vorlagendatei

Das InfoPath Formular setzt sich im Wesentlichen aus den XML-Dateien „manifest.xsf“, „myschema.xsd“, „template.xml“, „sampledata.xml“ und „view.xsl“ zusammen. Für jede

---

<sup>18</sup> Die Dateiendung „.cab“ (englisch: „cabinet“) beschreibt ein Format zur Komprimierung von Dateien.

## 5 Generierung des Formulars aus dem Modell

Ansicht (Formularseite) im Formular existiert eine eigene „view .xsl“-Datei. Werden externe Ressourcen, Datenverbindungen oder ähnliches im Formular verwendet, wird der Inhalt der Formularvorlagendatei entsprechenden XML-Dateien ergänzt.

Das Manifest bildet das Kernstück des Formulars. Es handelt sich um eine „Formulardefinitionsdatei“, welche Informationen über die Zusammensetzung eines Formulars enthält. In ihr werden alle zum Formular gehörenden Dateien angegeben sowie Inhalte und Eigenschaften der Formularvorlage definiert. Weiterhin werden im Manifest das Verhalten von Schaltflächen und Feldvalidierungen gesteuert. Verbindungen und Einstellungen zu SharePoint sind ebenfalls im Manifest definiert.

In der Schema-Datei „myschema.xsd“ wird das dazugehörige XML-Schema definiert, um die Daten des Formulars zu strukturieren. Es enthält Einschränkungen und Validierungen der Daten der Formularvorlage. Für jede Datenverbindung in einer Formularvorlage existieren eigene Schema-Dateien.

Die „sample.xml“ enthält Standardwerte für Formularfelder. Standardwerte werden ebenfalls in der „template.xml“-Datei (siehe Abb. 66) gespeichert. Die „template.xml“-Datei enthält zusätzlich noch eine „Processing Instruction“, d. h. eine Informationen, dass dieses Template mit einer entsprechenden Formularvorlage verbunden ist und dass das Formular in InfoPath geöffnet wird. Die XML-Datei, die beim Speichern des ausgefüllten Formulars erzeugt wird, entspricht der „template.xml“.



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?mso-infoPathSolution name="urn:schemas-microsoft-com:office:infopath:Form1:-myXSD-2013-04-02T09-18-23
3 <?mso-application progid="InfoPath.Document" versionProgid="InfoPath.Document.4"?>
4 <?mso-infoPath-file-attachment-present?>
5 <my:myFields xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:pc="http://schemas.microsoft.com
6 <my:Vorname></my:Vorname>
7 <my:GebDatum></my:GebDatum>
8 <my:Nachname></my:Nachname>
9 <my:Versicherungsnummer></my:Versicherungsnummer>
10 <my:Krankenkasse></my:Krankenkasse>
11 <my:Versicherungsart></my:Versicherungsart>
12 <my:Beschädigungsgrund></my:Beschädigungsgrund>
13 <my:Hersteller></my:Hersteller>
14 <my:FileAttachement xsi:nil="true"></my:FileAttachement>
15 <my:Vorgangsnummer></my:Vorgangsnummer>
16 </my:myFields>
17
```

Abbildung 66: Auszug aus template.xml

Für externe Datenverbindungen werden zusätzliche XML-Dateien von InfoPath erstellt. Im Brillenshop-Beispiel werden zum Beispiel die Auswahlmöglichkeiten des Feldes „Krankenkasse“ aus einer SharePoint-Liste herangezogen. In InfoPath bedeutet dies das Anle-

gen einer Datenverbindung zum Lesen von Daten aus SharePoint. Die Datenverbindung zum Auslesen der Krankenkassen-Liste wird durch fünf XML-Schema Dateien in der Formularvorlagendatei beschrieben. In der manifest.xsf wird ein entsprechendes DataObject-Element angelegt, welches Informationen zur Datenverbindung (wie zum Beispiel: wann Daten vom SharePoint abgefragt werden und welche Felder gelesen werden) enthält.

Eine Generierung des InfoPath Formulars aus dem Modell erfordert das Erstellen mehrerer XML-Dateien. Das InfoPath XML ist dabei sehr strikt und selbst kleinere Abweichungen im Aufbau der XML-Dateien führen zu Fehlermeldungen beim Öffnen des Formulars im InfoPath Designer. Von Microsoft InfoPath sind die einzelnen XML-Dateien und darin enthaltenen Elemente nicht sehr ausführlich dokumentiert.

### 5.2 Implementierung eines Formulargenerators

Es gilt, aus einem Modell eine entsprechende Exportschnittstelle zu InfoPath zu schaffen. Eine nicht unwesentliche Frage ist, ob für InfoPath eigenständige Modelle erstellt werden müssen, oder ob anhand eines Modells der Export zu einem cit Dialogassistenten und zum InfoPath Formular gewährleistet werden kann. Es gibt sicherlich Szenarien, in denen es durchaus denkbar wäre, nur ein Formulardatei zu erstellen und daraus entweder ein InfoPath bzw. cit Formular zu generieren. Die entscheidende Frage ist, inwieweit dies wirklich sinnvoll ist.

InfoPath und cit unterscheiden sich in einigen Punkten sehr voneinander. Während InfoPath weitestgehend ohne Programmcode auskommt, verwendet cit hauptsächlich Javascript Ausdrücke. Einige dieser Ausdrücke (zum Beispiel Zahlenformatierungen, reguläre Ausdrücke oder einfache Feldvergleiche) ließen sich mit relativ einfachen Aufwand von Javascript nach InfoPath programmatisch konvertieren. Dies hätte den Vorteil, dass die bereits vorhandene Modellierungsnotation weiter verwendet werden kann. Für eine komplexere Formularlogik ist dies nicht praktikabel. Im Gegensatz zu cit verwendet InfoPath Dialogmasken mit Auswahllisten, um Formatierungen, Regeln oder andere Einstellungen an den Formularansichten bzw. Formularfeldern vorzunehmen. Dies macht InfoPath benutzerfreundlicher, da aus vorhandenen Auswahllisten gewählt werden kann, und verringert Fehler durch falsche Schreibweisen. Es ließe sich jedoch ein Teil der Attribute (z. B. Pflichtfeld, Beschriftung, Gliederungsnummer) weiter nutzen.

### 5.2.1 Mapping der cit Modelle auf InfoPath

Eine weitere wichtige Frage ist, wie sich die cit Formularfelder auf entsprechende InfoPath Steuerelemente abbilden lassen. Der Composer folgt beim Erstellen des Dialogassistenten und dessen Eingabefeldern bestimmten Regeln. Zu den Regeln gehören unter anderem, wie die einzelnen Feldtypen letztendlich im fertigen Assistent dargestellt werden. Das Setzen von Multiplizitäten für wiederholende Abschnitte hat Auswirkungen auf die Darstellung im Formularassistenten. Es gilt, dieses Feldverhalten in InfoPath entsprechend abzubilden („mapping“). Das Mapping der Formularseiten und Eingabefelder ist, aufgrund der Unterschiede zwischen cit und InfoPath nicht ganz trivial. Cit unterscheidet im Modell zwischen mehreren Feldtypen (z. B. Zahlenfeld, Textfeld, Datumsfeld, Ja-/Nein-Feld, Auswahlfeld). InfoPath bildet einige dieser Feldtypen mit einem einzigen Feld, dem Textbox-Feld ab, andere hingegen werden in InfoPath voneinander getrennt (z. B. Checkbox, Combobox, Option Button in InfoPath; als Pendant zum Auswahlfeld im cit Modell). Bei einer Implementierung muss also vorher genau festgelegt werden, für welche Eingabefelder welche Steuerelemente verwendet werden. Auch hier kann wieder die Frage gestellt werden, ob es notwendig ist, zusätzliche Shapes zur Verfügung zu stellen oder ob die vorhandenen Shapes ausreichen und ob zusätzliche Attribute hinzuzufügen und das Mapping programmatisch vorzunehmen sind.

InfoPath weist einige Besonderheiten, insbesondere beim Ein- und Ausblenden von einzelnen Eingabefeldern auf einer Formularansicht, auf. Ein InfoPath Formular ist tabellarisch aufgebaut. Das BAföG-Formular enthält mehrere Spalten, eine für die Feldbeschriftung und eine für das Eingabefeld selbst. Soll ein Eingabefeld auf einer Formularansicht ausgeblendet werden, reicht es nicht aus, eine einzelne Regel am Eingabefeld zu erstellen, da sonst die Beschriftung und die Tabellenzeile, in der das Eingabefeld platziert ist, weiterhin angezeigt werden. Um eine ganze Tabellenzeile Ein- bzw. Auszublenden muss ein Container erzeugt werden. In InfoPath bedeutet dies das Verwenden des Section-Steuerlements. Die auszublendenden Steuerelemente sind innerhalb dieser Section zu platzieren. Per Regel kann anschließend die gesamte Section ausgeblendet werden. Es reicht also nicht nur aus, eine XML Struktur für das Eingabefeld zu erzeugen, sondern es müssen bei vorhandenen Abhängigkeiten zum Ein- oder Ausblenden zusätzliche XML-Elemente erzeugt werden, um das gewünschte Verhalten abzubilden.

Ein weiteres gutes Beispiel für die Unterschiede zwischen cit und InfoPath sind die Un-

terformulare in den cit Dialogassistenten. Die Unterformulare wurden in Abschnitt 3.1.3 bereits beschrieben. Solche Unterformulare wie im cit Assistenten lassen sich in InfoPath in dieser Art und Weise nicht ganz so einfach zu realisieren. Im BAföG-Formular mit InfoPath (siehe Anhang Seite XVII-XIX) wurde bewusst versucht, das cit Formular nachzubauen. Eine 1:1 Umsetzung des cit Dialogassistenten in InfoPath ist nicht immer sinnvoll und teilweise nur mit zusätzlichem Programmcode möglich bzw. unmöglich. Stattdessen muss ein praktikabler Weg gefunden werden, wie solche Inhalte in InfoPath dargestellt werden könnten. Im genannten Beispiel wird für jedes Kind ein eigenes Unterformular geöffnet. Mit normalen InfoPath Mitteln (ohne zusätzliche Programmierung) ist dies nicht möglich. Stattdessen wurde das Steuerelement „Repeating Section“ (wiederholender Abschnitt), verwendet, um das mehrfache Ausfüllen einer bestimmten Menge von Eingabefeldern zu ermöglichen.

Ein weiteres Beispiel sind Zahlen-, Maß- und Währungsfelder. Alle drei Eingabefelder sind Zahlenfelder. Währungsfelder enthalten zusätzlich eine Angabe zur Währung und Maßfelder Angaben zu Einheiten. Den Währungs- und Maßfeldern können mehrere Währungen bzw. Einheiten zugeordnet werden. Im Dialogassistenten erscheint ein entsprechendes Auswahlfeld, aus dem die Währung bzw. die Einheit ausgewählt werden kann. In einem InfoPath Formular lässt sich nur eine Währung angeben. Eine direkte Umsetzung in InfoPath in diesem speziellen Fall ist nicht möglich. Stattdessen müssen weitere Felder zur Auswahl der Währung bzw. Einheit angelegt werden.

Das Abbilden der Dialogseiten und Eingabefelder von cit auf InfoPath ist nicht immer trivial. An vielen Stellen machen sich die Besonderheiten beider Produkte bemerkbar. Sollen InfoPath Formulare generiert werden, gilt es einen für InfoPath geeigneten Weg zu finden.

### **5.2.2 Vorstellung des Prototyps**

Im Rahmen der Machbarkeitsanalyse wurde ein Prototyp entwickelt, mit dessen Hilfe einerseits die Machbarkeit einer Implementierung und andererseits der dafür benötigte Zeitaufwand evaluiert werden soll. Der Prototyp wurde mit dem Tool Microsoft Visual Studio 2010 in der Programmiersprache Visual Basic (VB) programmiert.

Eine technische Implementierung erfordert das Erstellen der XML-Dateien, welche in der

## 5 Generierung des Formulars aus dem Modell

Formularvorlagendatei enthalten sind. Aufgrund einer unzureichenden Dokumentation bezüglich der einzelnen XML-Dateien ist der Prototyp mit sehr viel „Reverse Engineering“ entstanden. „Reverse Engineering“ bedeutet, dass zunächst ein Formular in InfoPath mit verschiedenen Steuerelementen angelegt und anhand des Formulars die XML-Struktur mittels Programmcode nachkonstruiert wurde. Eine solche Herangehensweise ist sehr zeitaufwendig, aber notwendig, um ein funktionierendes InfoPath Formular zu erzeugen. Das Problem dabei ist, dass das InfoPath XML sehr strikt ist und keinerlei Abweichungen erlaubt, da sonst beim Öffnen des Formulars Fehler auftreten.

Der Prototyp wurde in die bestehende cit intelliForm Installation integriert. Zum Generieren des InfoPath Formulars aus dem Modell wurde zunächst auf eine Anpassung der Oberfläche und Erstellung von zusätzlichen Eingabemasken verzichtet. Es werden ausschließlich die vorhandenen cit Attribute genutzt. Der InfoPath Export wird mit dem Klick auf einen Button im Menüband gestartet (Abb. 67)

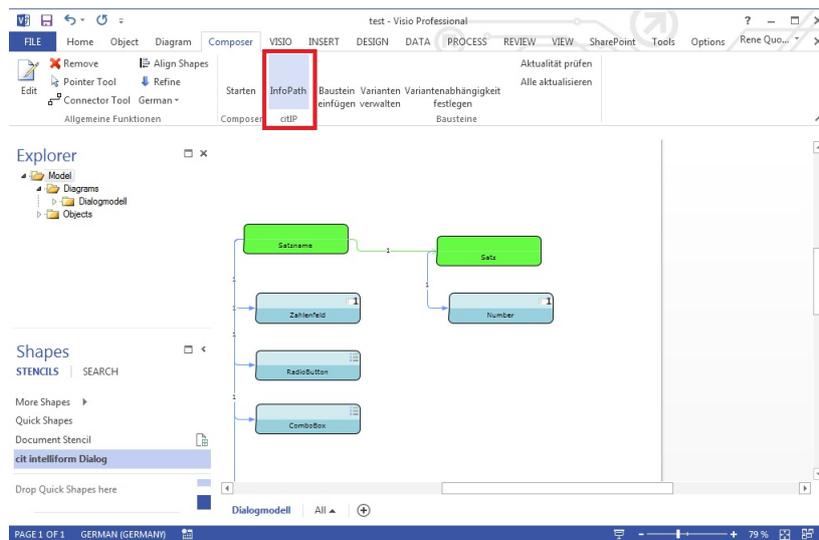


Abbildung 67: InfoPath - Export

Nach dem Klick auf die Schaltfläche im Menüband erfolgt zunächst eine Auswahl, wo die XML-Dateien gespeichert werden sollen. Anschließend wird der Programmcode ausgeführt und alle notwendigen XML-Dateien (manifest.xsf, sampledata.xml, template.xml, und die einzelnen view.xsl-Dateien angelegt). Nachfolgend wird grob auf die Funktionsweise und den Funktionsumfang des Prototyps eingegangen.

Der Prototyp besteht im Wesentlichen aus den fünf Prozeduren:

- Basefiles,

## 5 Generierung des Formulars aus dem Modell

- Objects,
- View,
- GetControl und
- CompFields.

Als erstes wird die Prozedur „Basefiles“ aufgerufen. Diese Prozedur dient dazu, die Grundstruktur der benötigten Dateien (manifest.xsf, sampledata.xml, template.xml und mySchema.xml) anzulegen. Die in diesen Dateien enthaltene Element-Struktur wird nach und nach erweitert. Anschließend erfolgt der Aufruf der Prozedur „Objects“. Sie stellt den Kern des Prototyps dar. Sie durchsucht das Modell zunächst nach den Satz-Shapes. Satz-Shapes stellen in der Regel eine Formularseite dar. Wird ein Satz-Shape als Composition modelliert, erscheinen die Eingabefelder des Satzes auf der Seite des übergeordneten Satzes als eigenständiger inhaltlicher Block. Zu jeder Formularseite muss eine view.xsl angelegt werden. Dazu wird die Prozedur „View“ aufgerufen. Sie erstellt eine entsprechende XSL-Datei für den Satz mit einer Grundstruktur. Diese Grundstruktur wird durch Hinzufügen von Steuerelementen nach und nach erweitert. Zu jedem Eingabefeld, das mit dem Satz verbunden wird, wird die Funktion „GetControl“ aufgerufen. Sie dient zur Konstruktion der XML-Elemente eines InfoPath Steuerelements. Für die einzelnen InfoPath Steuerelemente wurde eine Vorlagendatei (mapping.xml - Abb. 68) angelegt.



```
<?xml version="1.0" encoding="utf-8" ?>
<controls xmlns:pc="http://schemas.microsoft.com/office/infopath/2007/PartnerControls" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <!-- xd:binding; xsl:choose; xd:CtrlId-->
  <TextField>
    <span hideFocus="1" class="xdTextBox" title="" tabIndex="0" xd:xctname="PlainText" xd:binding="" xd:CtrlId="" style="WIDTH: 100%" xmlns:
      <xsl:value-of select="" />
    </span>
  </TextField>
  <Button>
    <div align="center">
      <input style="MARGIN: 1px" class="langFont" title="" value="" type="button" xd:CtrlId="" xd:xctname="Button" tabIndex="0" xmlns:xd="ht
    </div>
  </Button>
  <!-- Zahlenfeld Integer xd:binding; xsl:choose; xd:CtrlId -->
  <NumberField>
    <span hideFocus="1" class="xdTextBox xdBehavior_Formatting" title="" contentEditable="true" tabIndex="0" xd:xctname="PlainText" xd:bindi
      <xsl:attribute name="xd:num">
        <xsl:value-of select="" />
      </xsl:attribute>
      <xsl:choose>
        <xsl:when test="function-available('xdFormatting:formatString')">
          <xsl:value-of select="" />
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select="" />
        </xsl:otherwise>
      </xsl:choose>
    </span>
  </NumberField>
```

Abbildung 68: Mapping - InfoPath Steuerelemente

Sie enthält vordefinierte XML-Elemente, die die InfoPath Steuerelemente darstellen. Die-

se Vorlagen werden im Programmcode angepasst und erweitert. Die Erweiterung ist notwendig, um die Steuerelemente in InfoPath eindeutig voneinander zu unterscheiden und um bestimmte Eigenschaften des Eingabefeldes, wie zum Beispiel Feldformatierungen oder Auswahlmöglichkeiten für Auswahlfelder zu setzen. Ist das Eingabefeld zusammengesetzt, wird die Formulartabelle erweitert. Je nach Angabe am Attribut „Beschriftungsposition“ wird eine Beschriftung gesetzt und das Steuerelement in der entsprechenden Spalte der neu angelegten Tabellenzeile platziert. Zusätzlich zum Anlegen der Elemente in der view.xsl sind weitere Einträge in den Dateien mySchema.xsd, sampledata.xml und template.xml notwendig. Die Schema-Datei definiert den Dateityp des Steuerelements. Für Pflichtfelder werden im Manifest entsprechende Elemente für die Pflichtfeldüberprüfung angelegt.

Der Prototyp bietet eine Grundfunktionalität, die es erlaubt ein InfoPath Formular mit mehreren Ansichten und Formularfeldern zu erstellen. Mit dem Prototyp lassen sich Textfelder, Zahlenfelder, Auswahlfelder (Combo Box und Radio Button) und Infokästen aus dem cit Modell in InfoPath abbilden. Einige der cit Attribute werden bei der Erstellung des InfoPath Formulars berücksichtigt. Dazu gehören die Attribute Titel, Intro und Outro, Beschriftung, Beschriftungsposition, Gliederungsnummer (bei Eingabefeldern) und Pflichtfeld. Mit Hilfe dieser Attribute lassen sich bereits einfache InfoPath Formulare generieren. Was dem Prototyp fehlt, ist das Verwenden von SharePoint-Listen als externe Datenquelle für Auswahlfelder, eine Ablaufsteuerung, die den Benutzer Schritt für Schritt durch das Formular führt, sowie Hinweistexte für Fehleingaben. Die Umsetzung der Hinweistexte und der Ablaufsteuerung ist nicht ganz trivial. Hinter der Ablaufsteuerung der einzelnen Formularansicht verbirgt sich einiges an Logik. In InfoPath erfolgen der Ansichtenwechsel und das Absenden des Formulars an SharePoint über Schaltflächen. Soll verhindert werden, dass bei fehlenden Angaben oder Fehleingaben die Ansicht gewechselt und Hinweistexte angezeigt werden, erfordert dies das Erzeugen von mehreren Regeln an der Schaltfläche, am Eingabefeld sowie das Hinzufügen zusätzlicher Felder zum Anzeigen der Hinweistexte.

Als Ergebnis werden alle notwendigen XML-Dateien erzeugt. Die Formularvorlage lässt sich über die Manifest.xsf im InfoPath Designer öffnen. Um eine Formularvorlagendatei (.xsn) zu erhalten, kann es aus dem Designer heraus abgespeichert werden. Als Alternative dazu könnte die .xsn-Datei auch programmatisch erstellt werden. Dies erfordert das Komprimieren der einzelnen XML-Dateien zu einer cabinet-Datei. Um eine cabinet-Datei

zu erstellen, gibt es ein Windows Tool namens „Makecab“. Mit Hilfe dieses Tools lässt sich die .xsn-Datei auch programmatisch erzeugen.

### 5.2.3 Einschätzung des Implementierungsaufwandes

Eine Einschätzung des Implementierungsaufwandes ist schwierig und hängt von mehreren Faktoren ab. Der oben vorgestellte Prototyp ist während der Erstellung dieser Arbeit in einem Zeitraum von ca. 2-3 Monaten entstanden. Diese Zeit enthält ein gewisses Kennenlernen der Programmierumgebung (Visual Studio und VB.NET) sowie der Eigenheiten von IntelliForm und InfoPath. Die meiste Zeit wurde für das Abbilden der cit Elemente in InfoPath benötigt. Da die XML-Struktur von InfoPath nur spärlich dokumentiert ist, war es notwendig, den Aufbau der XML-Struktur von InfoPath genau zu analysieren, um die Formularseiten, Eingabefelder und Regeln entsprechend abzubilden.

Ein weiterer Faktor ist der Funktionsumfang. IntelliForm ist bereits seit ca. 5 Jahren auf den Markt und wurde seitdem stets weiterentwickelt. Somit hat sich auch der Funktionsumfang kontinuierlich erweitert. Dabei steckt sehr viel Aufwand im Detail. Es gilt herauszufinden, welche Kernfunktionalitäten benötigt werden und welche zusätzlichen Funktionalitäten in einer ersten Version enthalten sein sollen. Der Programmieraufwand der Kernfunktionalitäten, wie das Erstellen der einzelnen Ansichten mit den Eingabefeldern, der Ablaufsteuerung und dem Absenden an SharePoint, ist überschaubar.

Ein weiterer Zeitfaktor ist Gestaltung der Modellierungsoberfläche. InfoPath unterstützt den Benutzer an vielen Stellen beim Erstellen von Formularen mit Auswahlfeldern oder anderen Eingabemasken. Bei der Prototyp-Entwicklung wurde lediglich auf die cit Attribute zurückgegriffen. Es ist allerdings durchaus sinnvoll, weitere Masken zu entwerfen, die Auswahlfelder für bestimmte Steuerelementeigenschaften (z. B. Feldprüfung mit regulären Ausdrücken) bereithalten.

Der größte Aufwand besteht vor allem darin, die XML-Elemente anzulegen. Die Programmlogik selbst sollte für einen erfahrenen Programmierer in einen vertretbaren Aufwand zu bewältigen sein.

## 6 Fazit

Zusammenfassend lässt sich sagen, dass es viele Möglichkeiten gibt Webformulare zu gestalten. Die Generierung von Formularen anhand eines Modells zeigen durchaus Vorteile auf, die nicht nur den benötigten Zeitaufwand reduzieren, sondern gleichzeitig auch eine Trennung zwischen dem Inhalt des Formulars und dessen Darstellung mit sich bringt. Cit IntelliForm ist ein etabliertes Produkt im Bereich der öffentlichen Verwaltung, nicht nur zur Erstellung von Formularen sondern auch für die Abwicklung ganzer formularzentrischer Verwaltungsprozesse mit Hilfe computergestützter Systeme. Die Verabschiedung des E-Government Gesetzes stellt einen Meilenstein in der Modernisierung der öffentlichen Verwaltung dar. Zur Modernisierung gehören vor allem eine zunehmend elektronisch gestützte Datenverarbeitung und die elektronische Abwicklung von Verwaltungsprozessen. Microsoft SharePoint ist ein etabliertes Produkt, auch in der öffentlichen Verwaltung. Es ist daher notwendig, aktuelle Entwicklungen zu verfolgen und gegebenenfalls diese Richtung einzuschlagen. Viele Unternehmen und auch Behörden setzen bereits auf SharePoint und sehen die Möglichkeit, Geschäftsprozesse mit SharePoint zu unterstützen. SharePoint bietet eine Plattform zur Verwaltung von Dokumenten und zum kollaborativen Arbeiten sowie zur Abwicklung ganzer Geschäftsprozesse mit Hilfe von Workflows.

Cit IntelliForm verfolgt einen eigenen modellbasierten Ansatz zur Erstellung von Formularen. Es galt zu überprüfen, inwiefern sich der modellbasierte Ansatz von Cit in SharePoint integrieren lässt. Diese Arbeit hat gezeigt, dass sich mit Hilfe von InfoPath formularbasierte Anwendungen in SharePoint erstellen lassen. Weiterhin galt es zu überprüfen, ob sich InfoPath Formulare aus einem Modell generieren lassen und welche Vorteile sich daraus ergeben. Diese Arbeit hat gezeigt, dass eine Generierung von InfoPath Formularen durchaus machbar ist und einige Vorteile mit sich bringt. Eine Generierung der Formulare aus dem Modell spart nicht nur Zeit bei der Erstellung, sondern bringt deutlich bessere Möglichkeiten zur Modularisierung der Formulare mit sich. Somit lassen sich einzelne Bausteine in anderen Formularen wiederverwenden und sorgen so für eine gewisse Einheitlichkeit bei der Erfassung von Daten. Modelle bieten außerdem den Vorteil, dass sich daraus Informationen ablesen lassen, um entsprechende SharePoint Strukturen zu generieren. Eine andere Sicht ist das Ableiten des Formulars aus einer bestehenden SharePoint Struktur. Informationen aus SharePoint können genutzt, in ein Formularmodell importiert und mit zusätzlichen Eigenschaften versehen werden, um daraus ein Formular

zu generieren.

Eine endgültige Entscheidung zur Umsetzung einer Implementierung ist abhängig von zwei weiteren Faktoren. Ein Faktor ist der zu betreibende Aufwand für eine Implementierung. Der Aufwand hängt vor allem vom Funktionsumfang der Formulare ab. Cit inteliform ist ein ausgereiftes Produkt, welches seit ca. 5 Jahren auf dem Markt ist, stets weiterentwickelt wurde und wird. Der Funktionsumfang geht weit über die bloße Formularerstellung hinaus und unterstützt den Ablauf ganzer Verwaltungsprozesse. Eine ebenso umfangreiche Implementierung in der SharePoint-Welt ist mit viel Aufwand verbunden. Viele Szenarien lassen sich allerdings mit SharePoint Boardmitteln umsetzen, sodass mit Hilfe von InfoPath Formularen und SharePoint Workflows formularbasierte Anwendungen mit relativ kleinem Aufwand umgesetzt werden können. Eine Umsetzung von Kernfunktionalitäten bei der Generierung von InfoPath Formularen ist nicht all zu aufwendig. Die entscheidende Frage ist, wie viele Details sollen in einer ersten Version enthalten sein, und wie komplex die einzelnen Anwendungsszenarien wirklich sind.

Der zweite Faktor, der eine wichtige Rolle spielt, ist die Zukunftssicherheit von InfoPath. Die Versionen InfoPath 2010 und InfoPath 2013 sind sich sehr ähnlich. InfoPath 2013 enthält kaum Neuerungen zur Vorgängerversion, was bedeuten könnte, dass die Entwicklung von InfoPath bei Microsoft nicht mehr so stark vorangetrieben wird. Wird InfoPath tatsächlich zukünftig von Microsoft abgesetzt, bleibt abzuwarten welche Richtung Microsoft einschlägt. Weiterhin bleibt es abzuwarten, welche Möglichkeiten HTML5 zukünftig mit sich bringt. Nichts desto trotz kann aus dieser Arbeit entnommen werden, dass die modellbasierte Generierung von browserfähigen Formularen Vorteile mit sich bringt. Diese Vorteile gelten nicht nur gegenüber der Erstellung in InfoPath sondern auch gegenüber anderen Technologien die entweder auf eine programmatische Lösung oder auf einen grafischen Formulareditor setzen.

## Glossar

.NET	Das .NET Framework ist eine Entwicklungsplattform zum Entwickeln von windowsbasierten Anwendungen. Die Plattform unterstützt die Programmiersprachen C# und Visual Basic.
ActiveX	ActiveX ist eine Microsoft-Technologie für aktive Inhalte im Web. ActiveX Technologien sind für alle Windows Plattformen verfügbar. Die ActivX-Steuerelemente funktionieren deshalb standardmäßig nur im Microsoft Internet Explorer.
E-Government	‘E-Government bezeichnet die elektronische Abwicklung von Geschäftsprozessen der öffentlichen Verwaltung und Regierung und leistet einen wichtigen Beitrag zur Modernisierung der Verwaltung: Die Prozesse werden beschleunigt, die Kommunikation vereinfacht und damit insgesamt die Qualität und Effizienz des öffentlichen Verwaltungshandelns erhöht...’ [Bun13a].
elektronische Signatur	Elektronische Signaturen sind ‘Daten in elektronischer Form, die anderen elektronischen Daten beigefügt oder logisch mit ihnen verknüpft sind und die zur Authentifizierung dienen...’ §2 Nr. 3 SigG [Bun01].
InfoPath	Microsoft InfoPath ist ein Anwendungsprogramm zum Erstellen und Ausfüllen von XML-basierten Formularen. Ab der Version 2007 wird zwischen dem InfoPath Designer und dem InfoPath Filler unterschieden. InfoPath Formulare lassen sich mit Hilfe der InfoPath Forms Services auch in einem Browser ausfüllen.
InfoPath Filler	InfoPath Filler ist seit der Version 2007 Bestandteil von Microsoft InfoPath. Mit der Clientanwendung können im InfoPath Designer erstellte Formulare ausgefüllt werden.

Java/JBOSS	Java ist eine Laufzeitumgebung und Programmiersprache. Sie wurde im Jahr 1995 von Sun Microsystems veröffentlicht und bildet heute die Grundlage vieler moderner Anwendungen und Websites. JBOSS (auch JBOSS Application Server) ist ein Java Enterprise Edition kompatibler Anwendungsserver. Ein Java Anwendungsserver stellt standardisierte Kernkomponenten zur Ausführung von Java-Anwendungen bereit [McC09, S. 4f].
JSP	Java Server Pages ist eine Web-Programmiersprache. Mit Hilfe von Java Server Pages lassen sich dynamische XML- und HTML-Ausgaben auf einem Webserver erzeugen.
Makecab	Makecab.exe ist ein Microsoft Dienstprogramm zum Komprimieren von Dateien. Es wird mit Microsoft Visual Basic installiert. Das Dienstprogramm wird über MS-DOS Eingabeaufforderung aufgerufen.
Microsoft Visual Studio	Microsoft Visual Studio ist eine von Microsoft entwickelte Entwicklungsumgebung. Sie unterstützt die Entwicklung von Windows Anwendungen und Webseiten.
SharePoint	SharePoint ist ein Unternehmensportal für Kommunikations- und Informationsaustausch der Firma Microsoft.
SharePoint Designer	Der SharePoint Designer ist ein Tool zur Entwicklung von SharePoint-Anwendungen. Der Designer ermöglicht fortgeschrittenen Benutzern und Entwicklern die Entwicklung einer codefreier SharePoint-Lösungen.
VB.NET	Visual Basic .NET (VB.NET) ist eine auf dem Microsoft .NET Framework aufbauende objektorientierte Programmiersprache.
Webservice	Ein Webservice ist eine auf ein Websystem bereitgestellte Softwarekomponente. Der Webservice (oder auch Webdienst) kann in beliebigen Anwendungen verwendet werden.

Workflow	Ein Workflow ist eine logisch und zeitlich definierte Abfolge von abteilungsübergreifenden, fachlich zusammenhängenden Arbeitsschritten. Workflows unterstützen die computergestützte Abwicklung von Geschäftsprozessen.
Workflow Foundation	‘Die Windows Workflow Foundation ist eine Bibliothek innerhalb des Microsoft .NET Frameworks, die einen komplexen Arbeitsablauf (Workflow) als automatisiertes Programm abbildet. Die Logik des Geschäftsprozesses und die technische Entwicklungslogik werden dabei stets voneinander getrennt. Grafisch wird der logische Ablauf durch ein Ablaufdiagramm veranschaulicht.’ [Nox13].
XAML	Extensible Application Markup Language (XAML) ist eine deklarative Auszeichnungssprache zur Gestaltung von Oberflächen in .NET Anwendungen und zum Entwickeln von Workflows. XAML basiert auf XML.
XML	Extensible Markup Language (XML) ist eine Auszeichnungssprache, die eine layoutunabhängige Beschreibung von Dokumenten ermöglicht. XML ist als universelles Datenaustauschformat konzipiert worden [Bec09, S. 3].

## Literatur

- [And13] Andreas Pidde. Web-Dokument. Website, 08 2013.  
<http://userpages.uni-koblenz.de/~pidde/lupe/document.html>, Zugriff am 12.08.2013.
- [Bec09] Margit Becher. *XML*. W3I-Verlag, Witten, 2009.
- [Bun01] Bundesministerium der Justiz. Signaturgesetz. Website, 05 2001.  
<http://www.itwissen.info/definition/lexikon/accessibility-Zugaenglichkeit.html>, Zugriff am 03.11.2013.
- [Bun02] Bundesverwaltungsamt - Bundesstelle für Büroorganisation und Bürotechnik (BBB). BBB-Arbeitshandbuch - Bürgernahe Verwaltungssprache. PDF, 4. Auflage 2002.
- [Bun13a] Bundesministerium des Innern. E-Government. Website, 08 2013.  
[http://www.cio.bund.de/DE/Strategische-Themen/E-Government/egovernment\\_node.html](http://www.cio.bund.de/DE/Strategische-Themen/E-Government/egovernment_node.html), Zugriff am 12.08.2013.
- [Bun13b] Bundesministerium des Innern. E-Government-Gesetz. Website, 08 2013.  
[http://www.cio.bund.de/DE/Strategische-Themen/E-Government/E-Government-Gesetz/egovernment\\_gesetz\\_node.html](http://www.cio.bund.de/DE/Strategische-Themen/E-Government/E-Government-Gesetz/egovernment_gesetz_node.html), Zugriff am 12.08.2013.
- [cit13a] cit. 20 Jahre cit. PowerPoint Präsentation, 10 2013.
- [cit13b] cit. cit intelliForm Assistants - Handbuch. PDF, 03 2013.
- [cit13c] cit. cit intelliForm Composer für Formular-Assistenten - Handbuch. PDF, 03 2013.
- [CJ09] G. Gaffney C. Jarret. *Forms that Work*. Morgan Kaufmann, Burlington, 2009.
- [Dec12] Deckert, Füllgraf, Quoos. Vergleich von BPMN-Tools zur Modellierung von Geschäftsprozessen. PDF, 07 2012.  
<http://opus4.kobv.de/opus4-fhbrb/files/76/20120707-FHB-BPMN-Tools.pdf>, Zugriff am 12.08.2013.
- [Dub03] M. Dubinko. *XForms Essentials*. O'Reilly, Sebastopol, 2003.

- [ITW13a] ITWissen. Stichwort: Formular. Website, 08 2013.  
<http://www.itwissen.info/definition/lexikon/form-Formular.html>, Zugriff am 12.08.2013.
- [ITW13b] ITWissen. Stichwort: Scriptsprache. Website, 08 2013.  
<http://www.itwissen.info/definition/lexikon/Scriptsprache-script-1anguaage.html>, Zugriff am 19.08.2013.
- [ITW13c] ITWissen. Stichwort: Workflow. Website, 08 2013.  
<http://www.itwissen.info/definition/lexikon/Workflow-workflow.html>, Zugriff am 12.08.2013.
- [Kra07] H. Krallmann. *Systemanalyse im Unternehmen*. Oldenburg Verlag, 5. Auflage, 2007.
- [McC09] M. McCullough. *JBOSS im Einsatz*. Carl Hanser Verlag, München, 2009.
- [Mic13a] Microsoft. Eine Einführung in Formulare und Formularvorlagen für Anfänger. Website, 08 2013.  
<http://office.microsoft.com/de-de/infopath-help/eine-einfuehrung-in-formulare-und-formularvorlagen-fur-anfanger-HA001155963.aspx>, Zugriff am 09.08.2013.
- [Mic13b] Microsoft. Einführung in Workflows. Website, 08 2013.  
<http://office.microsoft.com/de-de/sharepoint-server-help/einfuehrung-in-workflows-HA010154424.aspx>, Zugriff am 12.08.2013.
- [Mic13c] Microsoft. InfoPath help - Design a template part to reuse in multiple form templates. Website, 10 2013.  
<http://office.microsoft.com/en-us/infopath-help/design-a-template-part-to-reuse-in-multiple-form-templates-HA010150746.aspx>, Zugriff am 12.08.2013.
- [Nox13] Noxum. Windows Workflow Foundation. Website, 11 2013.  
<http://www.noxum.com/de/produkte/workflow-service/wissenswertes/windows-workflow-foundation/>, Zugriff am 03.11.2013.
- [Ram04] T. V. Raman. *XForms: XML Powered Web Forms*. Pearson Education, 2004.
- [Ver09] Verwaltungsakademie Berlin. Verwaltungstechnik Bund. PDF, 2. Ausgabe 2009.

[http://www.berlin.de/verwaltungsakademie/dokumente/pdf/lehrbriefe/2009\\_Verwaltungstechnik\\_Bund\\_mit-ISBN.pdf](http://www.berlin.de/verwaltungsakademie/dokumente/pdf/lehrbriefe/2009_Verwaltungstechnik_Bund_mit-ISBN.pdf), Zugriff am 12.08.2013.

# Anhang

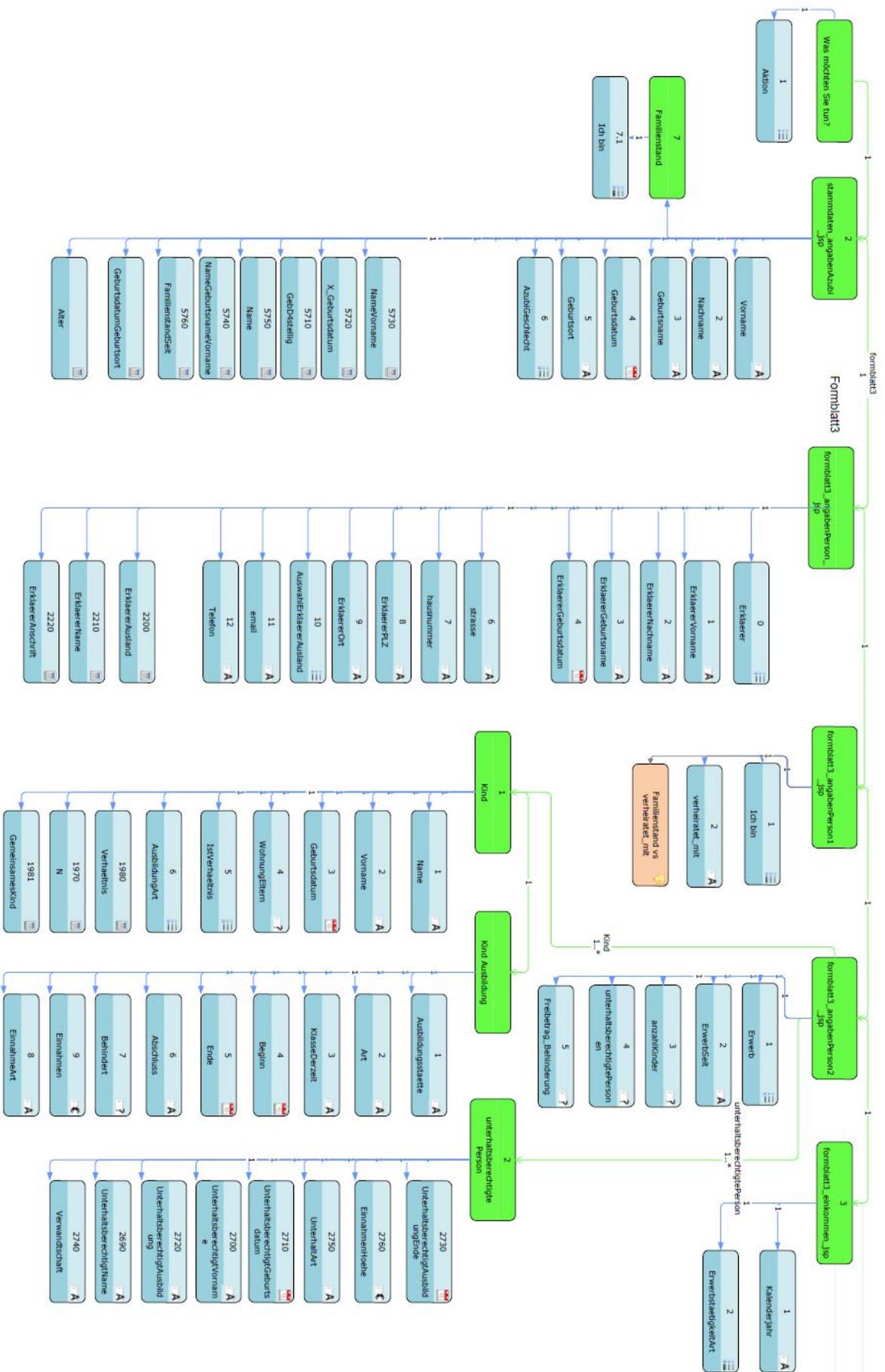


Abbildung 69: Ausschnitt BaföG-Modell Teil 1

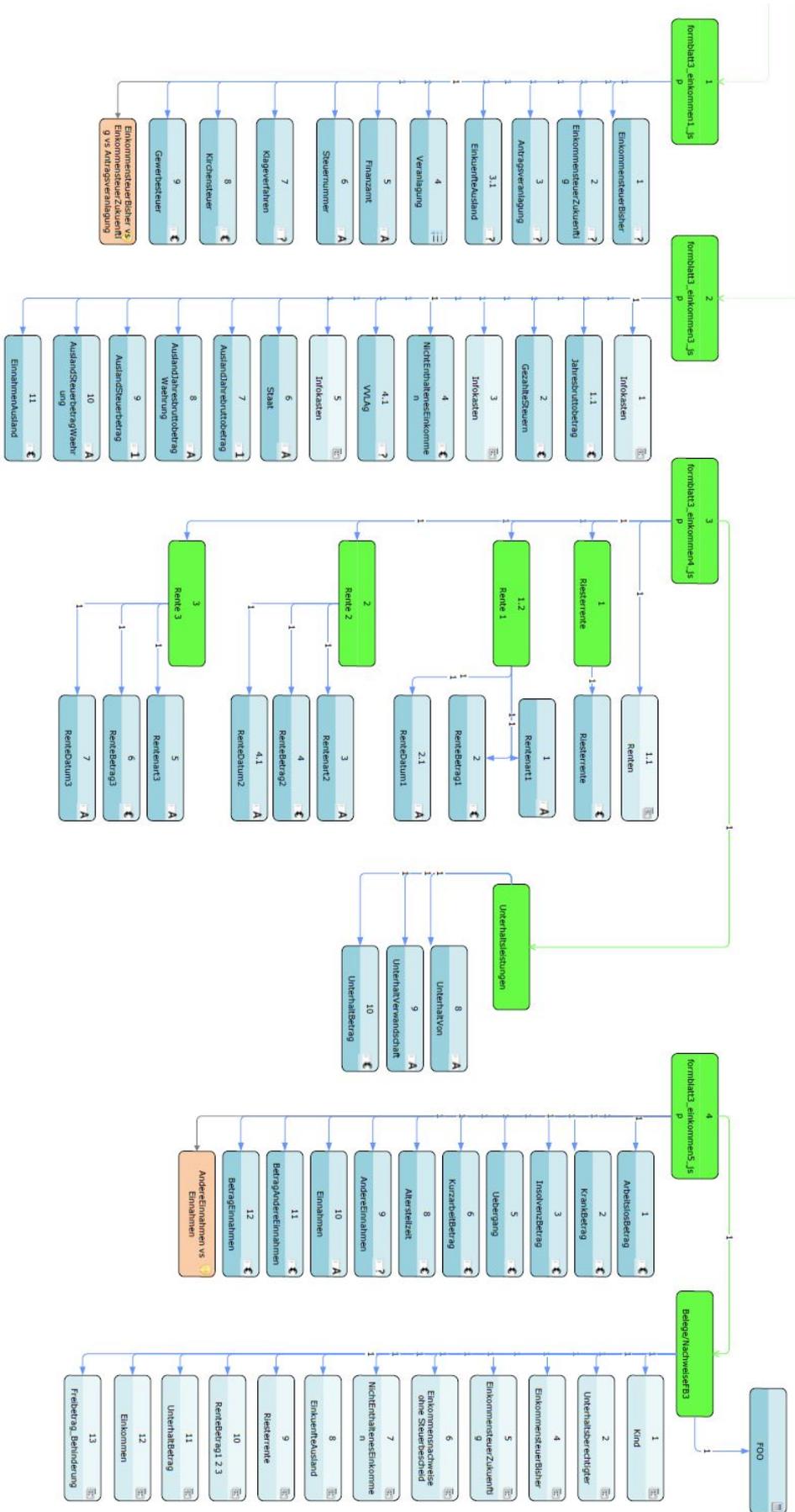


Abbildung 70: Ausschnitt BaföG-Modell Teil 2

### Bafög Online

**Was möchten Sie tun?**

Was möchten Sie tun?\*

- Ich bin Auszubildende/-r und möchte meinen Antrag auf Ausbildungsförderung stellen
- Ich bin Vater / Mutter / Ehegatte oder eingetragener Lebenspartner eines Auszubildenden und möchte meine Erklärung zum Antrag einreichen (Formblatt 3)
- Ich möchte einzelne Formulare zur Ausbildungsförderung online ausfüllen

Die mit Stern(\*) markierten Felder sind Pflichtfelder

Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

Abbildung 71: InfoPath - BAföG-Formular I

### Bafög Online

**Personenbezogene Angaben zum Auszubildenden**

Vorname:  \*

Nachname:  \*

Geburtsname:

Geburtsdatum:  \*

Geburtsort:  \*

Die mit Stern(\*) markierten Felder sind Pflichtfelder

Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

Abbildung 72: InfoPath - BAföG-Formular II

### Bafög Online

**Angaben zur eigenen Person**

Angaben beziehen sich auf den Bewilligungszeitraum

Erklärung:\*

- des Ehegatten/eingetr. Lebenspartner
- des Vaters
- der Mutter

*Bei einem Elternteil ohne eigenes Einkommen (auch keinen Minijob) reicht es aus, wenn auf dem Formblatt 3 des anderen Elternteils die Zusatzklärung unterschrieben wird.*

Vorname:  \*

Nachname:  \*

Geburtsname:  \*

Geburtsdatum:  \*

Straße:  \*

Hausnummer:  \*

PLZ:  \*

Ort:  \*

Staat:  \*

E-Mail:  \*

*(Optional: Diese Eingabe erleichtert uns die Kommunikation mit Ihnen.)*

Telefon:  \*

*(Optional: Diese Eingabe erleichtert uns die Kommunikation mit Ihnen.)*

Die mit Stern(\*) markierten Felder sind Pflichtfelder

Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

Abbildung 73: InfoPath - BAföG-Formular III

**BaföG Online**

**Angaben zur eigenen Person**

Ich bin:\*

ledig

verheiratet/eingetragener Lebenspartner

dauernd getrennt lebend

verwitwet

geschieden

verheiratet/in eingetragener Lebenspartnerschaft mit:

Zurück Weiter

Die mit Stern(\*) markierten Felder sind Pflichtfelder

Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

Abbildung 74: InfoPath - BAföG-Formular IV

**BaföG Online**

**Angaben zur eigenen Person**

Erwerb:\*

Arbeiter

Angestellter

Beamter

selbständig

nicht mehr erwerbstätig

seit (MM/JJJJ):

Haben Sie Kinder (ohne Antragsteller/in), welche von Ihnen unterhalten werden, sich in Ausbildung befinden oder in Ihren Haushalt aufgenommen wurden:\*

Ja  Nein

Gibt es Ihnen gegenüber unterhaltsberechtigte Person/en (z.B. dauernd getrennt lebender/geschiedener Ehegatte, zweiter Ehegatte, Eltern)?\*

Ja  Nein

Möchten Sie für sich, die Antragstellerin/den Antragsteller oder eine andere Ihnen gegenüber unterhaltsberechtigte Person einen Freibetrag/Freibeträge wegen einer anerkannten Behinderung beantragen?\*

Ja  Nein

Zurück Weiter

Die mit Stern(\*) markierten Felder sind Pflichtfelder

Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

Abbildung 75: InfoPath - BAföG-Formular V

**BaföG Online**

**Angaben zum Kind**

Bitte machen Sie hier Angaben zu/m Kind/ern (ohne Antragsteller/in), soweit sie von Ihnen unterhalten werden, sich in Ausbildung befinden befinden oder in Ihren Haushalt aufgenommen wurden.

Klicken Sie auf "Kind hinzufügen", um das zugehörige Unterformular auszufüllen und hier als Eintrag hinzuzufügen. Wenn Sie mit diesem Formularbereich fertig sind, klicken Sie auf "Weiter".

Keine Einträge vorhanden. Klicken Sie auf "Kind hinzufügen", um den ersten Eintrag hinzuzufügen.

Zurück Weiter

Die mit Stern(\*) markierten Felder sind Pflichtfelder

Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

Abbildung 76: InfoPath - BAföG-Formular VI

Bafög Online

**Angaben zum Kind: Allgemeines**

Name:  \*

Vorname:  \*

Geburtsdatum:  \*

Wohnung bei den Eltern: \*  Ja  Nein

Verhältnis:\*

Gemeinsames Kind der Eltern der/des Auszubildenden/oder gemeinsames Kind der/des Auszubildenden und des Ehegatten

Kind nur im Verhältnis zum Vater der/des Auszubildenden

Kind nur im Verhältnis zur Mutter der/des Auszubildenden

Kind nur im Verhältnis zum Ehegatten der/des Auszubildenden

Stiefkind, Pflegekind, Enkelkind soweit in Ihrem Haushalt aufgenommen

Das Kind:\*

geht zur Schule/studiert/befindet sich in Ausbildung

ist nicht in Ausbildung aber unterhaltsberechtig (auch nicht schulpflichtige Kinder)

Kind hinzufügen

Abbrechen
Weiter

Die mit Stern(\*) markierten Felder sind Pflichtfelder

Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

Abbildung 77: InfoPath - BAföG-Formular VII

Bafög Online

**Angaben zum Kind: Ausbildung**

*Kind: Quoos, René*

Name der Ausbildungsstätte:

Art des Ausbildungsverhältnisses:

derzeitige/s Klasse/Semester:

Ausbildungsbeginn:

voraussichtliches Ausbildungsende:

voraussichtlicher Abschluss als:

Ausbildungsmaßnahmen zur beruflichen Förderung als behinderter Mensch:  Ja  Nein

Art der Einnahme/n (Belege bitte in Kopie beifügen):  \*

Einnahmen monatlich:  \*

Abbrechen
Weiter

Die mit Stern(\*) markierten Felder sind Pflichtfelder

Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

Abbildung 78: InfoPath - BAföG-Formular VIII

**Bafög Online**

**Angaben zum Kind**

Bitte machen Sie hier Angaben zu/m Kind/ern (ohne Antragsteller/in), soweit sie von Ihnen unterhalten werden, sich in Ausbildung befinden oder in Ihren Haushalt aufgenommen wurden.

Die mit Stern(\*) markierten Felder sind Pflichtfelder

Klicken Sie auf "Kind hinzufügen", um das zugehörige Unterformular auszufüllen und hier als Eintrag hinzuzufügen. Wenn Sie mit diesem Formularbereich fertig sind, klicken Sie auf "Weiter".

Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

Folgende Einträge sind vorhanden:

Vorname	Nachname
Quoos	René

Abbildung 79: InfoPath - BAföG-Formular IX

**Bafög Online**

**Einkommen**

Für alle nachfolgenden Fragen sind die Verhältnisse **im vorletzten Kalenderjahr** vor Beginn des Bewilligungszeitraumes (BWZ) maßgebend,

Die mit Stern(\*) markierten Felder sind Pflichtfelder

also die des Kalenderjahres:  \*

Art der Erwerbstätigkeit:\*

- erwerbstätig als rentenversicherungspflichtige/r Arbeitnehmer/in (z.B. Arbeiter/in, Angestellte/r) oder in Ausbildung
- erwerbstätig als nichtrentenversicherungspflichtige/r Arbeitnehmer/in (z.B. Beamtin/Beamter oder Beamtin/Beamter im Ruhestand)
- erwerbstätig als Nichtarbeitnehmer/in (z.B. Selbstständige/r) oder auf Antrag von der Rentenversicherungspflicht befreite/r oder wegen geringfügiger Beschäftigung versicherungsfreie/r Arbeitnehmer/in
- nicht erwerbstätig oder im Ruhestandsalter nicht erwerbstätig (z.B. Altersrentner/in)

Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

Abbildung 80: InfoPath - BAföG-Formular X

BaföG Online

**Einkommen**

Wurden Sie für das entsprechende Kalenderjahr zur Einkommensteuer veranlagt (Bescheid in Kopie beifügen)?\*

Ja  Nein

Wurden Sie noch für das entsprechende Kalenderjahr zur Einkommensteuer veranlagt (Bescheid nach Erhalt in Kopie übersenden)?\*

Ja  Nein

Erfolgte eine Antragsveranlagung nach § 46 Abs. 2 Nr. 8 EStG (früherer Lohnsteuerjahresausgleich) (Bescheid in Kopie beifügen)?\*

Ja  Nein

**Wurden Einkünfte im Ausland erzielt?\***

Ja  Nein

Die Veranlagung erfolgt/e zusammen mit:

dem derzeitigen Ehegatten

dem Vater der/des Auszubildenden

der Mutter der/des Auszubildenden

Beim Finanzamt:

Steuer-Nr.:

Ist zu dem beigefügten Einkommensteuerbescheid ein Einspruchs-/Klageverfahren anhängig?

Ja  Nein

Angabe zur Kirchensteuer, soweit nicht im Steuerbescheid enthalten (Bescheid in Kopie beifügen):  EUR

Angaben zur Gewerbesteuer (Bescheid vollständig in Kopie beifügen):  EUR

Die mit Stern(\*) markierten Felder sind Pflichtfelder

Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

Abbildung 81: InfoPath - BAföG-Formular XI

BaföG Online

**Einkommen**

Wenn keine Veranlagung zur Einkommensteuer durchgeführt worden ist und auch nicht mehr durchgeführt wird.

Jahresbruttobetrag der Einnahmen:  EUR  
(Nachweise in Kopie beifügen)

hierauf gezahlte/abgeführte Steuern:  EUR  
(Nachweise in Kopie beifügen)

Wenn Einnahmen im Steuerbescheid nicht enthalten sind (z.B. wegen Geringfügigkeit vom Arbeitgeber pauschal versteuerte Einnahmen).

Jahressumme: \* EUR  
(Nachweise in Kopie beifügen)

Es wurden vom Arbeitgeber vermögenswirksame Leistungen erbracht:\*

Ja  Nein

Die mit Stern(\*) markierten Felder sind Pflichtfelder

Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

Abbildung 82: InfoPath - BAföG-Formular XII

## BaföG Online

**Renten**

**Riesterrente**  
Bescheinigung nach §92 Einkommensteuergesetz ist beizufügen.

Wenn von Ihnen geförderte Altersvorsorgebeiträge nach §82 Einkommensteuergesetz (Beiträge zur "Riester-Rente") gezahlt wurden:

EUR

Die mit Stern(\*) markierten Felder sind Pflichtfelder

Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

Sofern Sie im Kalenderjahr keine Rente bezogen haben füllen Sie die "Brutto-Jahressumme" mit "0" aus. Es sind auch private Renten oder Unfallrenten anzugeben.

**Rente 1**  
Rentenbescheid, aus dem die **Bruttorente** ersichtlich ist in Kopie beifügen.

Art der Rente:

Brutto-Jahressumme: \* EUR

Rentenbeginn:

**Rente 1**  
Rentenbescheid, aus dem die **Bruttorente** ersichtlich ist in Kopie beifügen.

Art der Rente:

Brutto-Jahressumme: \* EUR

Rentenbeginn:

**Rente 1**  
Rentenbescheid, aus dem die **Bruttorente** ersichtlich ist in Kopie beifügen.

Art der Rente:

Brutto-Jahressumme: \* EUR

Rentenbeginn:

Abbildung 83: InfoPath - BAföG-Formular XIII

## BaföG Online

**Unterhaltsleistungen**  
Gemeint sind Unterhaltsleistungen die Sie erhalten.

Unterhaltsleistungen von:

Verwandtschaftsverhältnis zum/zur Erklärenden:

Jahressumme: \* EUR

Die mit Stern(\*) markierten Felder sind Pflichtfelder

Pflichtfelder, die auf Sie nicht zutreffen sind mit 0 auszufüllen

Abbildung 84: InfoPath - BAföG-Formular XIV

## Bafög Online

**Einkommen nach der BAföG-Einkommensverordnung**  
diese finden Sie [hier](#).

Arbeitslosengelt/ Unerhaltsgeld (Jahressumme):	<input style="width: 90%;" type="text"/> * EUR
Krankengeld (Netto- Jahressumme):	<input style="width: 90%;" type="text"/> * EUR
Insolvenzgeld (Jahressumme):	<input style="width: 90%;" type="text"/> * EUR
Übergangsgeld (Jahressumme):	<input style="width: 90%;" type="text"/> * EUR
Kurzarbeitergeld (Jahressumme):	<input style="width: 90%;" type="text"/> * EUR
Aufstockungsbeträge nach dem Altersteilzeitgesetz bzw. vergleichbare Leistungen nach beamtenrechtlichen Grundsätzen oder Betriebsrenten:	<input style="width: 90%;" type="text"/> * EUR
Ich hatte anderen Einnahmen nach der BAföG-Einkommens- verordnung:*	<input type="radio"/> Ja <input type="radio"/> Nein
und zwar (Nachweise in Kopie beifügen):	<input style="width: 90%;" type="text"/>
Jahressumme:	<input style="width: 90%;" type="text"/> * EUR
Weitere Einnahmen, soweit noch nicht aufgeführt Jahressumme (Nachweise in Kopie beifügen):	<input style="width: 90%;" type="text"/> * EUR

Die mit Stern(\*)  
markierten Felder  
sind Pflichtfelder

Pflichtfelder, die auf  
Sie nicht zutreffen  
sind mit 0  
auszufüllen

Abbildung 85: InfoPath - BAföG-Formular XV

## Bafög Online

**Belege/Nachweise**

- Nachweise weitere Kinder/Pflegekinder/Stiefkinder (ohne Antragsteller)  
z.B.: Schulbescheinigung (ab Klasse 9), Ausbildungsvertrag, Vertrag über FSJ/BFD,  
Einkommensnachweise, etc.
- Vollständiger Bescheid über Einkommenssteuer für das Jahr 2010 (auch die Erläuterungen):
- Bescheinigung über die Einkünfte die nicht im Steuerbescheid enthalten sind
- Bescheinigung nach § 92 Einkommensteuergesetz.
- Rentenbescheid, aus dem die Bruttorente ersichtlich ist in Kopie beifügen.
- Nachweis über Unterhaltsleistungen.
- Nachweise über die Einnahmen nach der BAföG-Einkommensverordnung (z.B.: Arbeitslosengeld,  
Kurzarbeitergeld, Krankengeld, Altersteilzeitgeld)(BAföG Einkommensverordnung)

Die mit Stern(\*)  
markierten Felder  
sind Pflichtfelder

Pflichtfelder, die auf  
Sie nicht zutreffen  
sind mit 0  
auszufüllen

Abbildung 86: InfoPath - BAföG-Formular XVI

# Ehrenwörtliche Erklärung

## Erklärung

Ich versichere eidesstattlich, dass die vorliegende Arbeit mit dem Titel

Modellbasierte Generierung dynamischer Webformulare

von mir selbstständig, ohne Hilfe Dritter und ausschließlich unter Verwendung der angegebenen Quellen angefertigt wurde. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen sind, habe ich als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form, auch nicht in Teilen, keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Brandenburg, den 07.11.2013

Unterschrift des Kandidaten