

Michael Buck

Untersuchung zum Einsatz von XML-Editoren als  
WYSIWYG-Tools zur XML-Templateerstellung,  
prototypisch am Beispiel einer softwaregestützten  
Entscheidungsvorlage für Darlehen

BACHELORARBEIT

HOCHSCHULE MITTWEIDA

---

UNIVERSITY OF APPLIED SCIENCES

Fakultät Mathematik/Physik/Informatik

Mittweida, 2009

Michael Buck

Untersuchung zum Einsatz von XML-Editoren als  
WYSIWYG-Tools zur XML-Templateerstellung,  
prototypisch am Beispiel einer softwaregestützten  
Entscheidungsvorlage für Darlehen

eingereicht als

BACHELORARBEIT

an der

HOCHSCHULE MITTWEIDA

---

UNIVERSITY OF APPLIED SCIENCES

Fakultät Mathematik/Physik/Informatik

Dresden, 2009

Erstprüfer: Prof. Dr.-Ing. Wilfried Schubert  
Zweitprüfer: Dipl.-Inf. Andreas Haake

Vorgelegte Arbeit wurde verteidigt am:

# Inhaltsverzeichnis

<b>Bibliografische Beschreibung .....</b>	<b>III</b>
<b>Abkürzungsverzeichnis .....</b>	<b>IV</b>
<b>Abbildungsverzeichnis.....</b>	<b>V</b>
<b>Tabellenverzeichnis .....</b>	<b>V</b>
<b>1 Einleitung.....</b>	<b>1</b>
1.1 Motivation.....	1
1.2 Voraussetzungen .....	2
1.3 Aufgabenstellung und Ziele .....	2
1.4 Aufbau der Arbeit .....	3
<b>2 Theoretische Grundlagen .....</b>	<b>4</b>
2.1 XML-Allgemein.....	4
2.1.1 XML in der Praxis.....	6
2.1.2 Klassifizierung von XML-Dokumenten .....	8
2.1.3 Abbildung von Daten zwischen XML-Dokumenten und relationalen Datenbanken.....	9
2.1.4 Erstellung von XML-Templates für ProSAB .....	10
2.2 Zusammenwirken von XML-Templates und JasperReports .....	11
2.2.1 Erzeugung von Dokumenten mittels ProSAB und JasperReports .....	12
2.2.2 Antragsbearbeitungs-Software ProSAB.....	14
2.2.3 Reporting-Tool JasperReports .....	15
2.2.4 Struktur eines ProSAB-Templates .....	15
2.3 XML-Editoren als WYSIWYG-Tools.....	18
2.3.1 Allgemeiner Aufbau von XML-Editoren.....	19
2.3.2 Verwendung der WYSIWYG-Layout-Ansicht bei XML-Editoren.....	22
2.4 Lösungsansatz zur Untersuchung der XML-Editoren .....	23
<b>3 Evaluation von WYSIWYG-fähigen XML-Editoren .....</b>	<b>24</b>
3.1 Konzeption .....	24
3.1.1 Ist-Analyse .....	24
3.1.2 Soll-Konzeption .....	25
3.1.3 Aufstellung von Bewertungskriterien .....	26
3.1.4 Erhebung der Produktinformationen.....	32
3.2 Vorauswahl .....	33
3.2.1 Ausgewählte XML-Editoren.....	34
3.2.2 Aussortierte XML-Editoren .....	37
3.3 Einsatzbewertung der ausgewählten XML-Editoren .....	39
3.3.1 Installation und Test.....	39
3.3.2 Detailbewertung anhand der Bewertungskriterien.....	39
3.3.3 Auswahl des XML-Editors .....	50
<b>4 Prototypische Realisierung unter Anwendung von Oxygen .....</b>	<b>52</b>
4.1 Was ist eine Entscheidungsvorlage.....	52
4.2 Dokumentaufbau der Entscheidungsvorlage für Darlehen .....	53
4.3 ProSAB- und JasperReports-Bestandteile des Templates Entscheidungsvorlage.....	55
4.4 Erstellen des Templates Entscheidungsvorlage mit dem ausgewählten XML-Editor .....	58
4.5 Test des Templates „EEV für Darlehen“ in ProSAB.....	61

4.6 Erfahrungen beim Einsatz WYSIWYG-fähiger XML-Editoren .....	63
<b>5 Zusammenfassung und Ausblick.....</b>	<b>66</b>
5.1 Erreichte Ergebnisse .....	66
5.2 Ausblick .....	67

**Glossar**

**Literaturverzeichnis**

**Anhang**

## **Bibliografische Beschreibung**

Buck, Michael:

Untersuchung zum Einsatz von XML-Editoren als WYSIWYG-Tools zur XML-Templateerstellung, prototypisch am Beispiel einer softwaregestützten Entscheidungsvorlage für Darlehen. - 67 S., Mittweida: Hochschule Mittweida, Fakultät Mathematik/Physik/Informatik, Bachelorarbeit, 2009

Referat:

Die Verwendung von grafischen Editoren zur Erstellung von XML-Dokumenten bietet zahlreiche Neuerungen gegenüber herkömmlichen XML-Editoren. Die WYSIWYG-Layout-Ansicht ermöglicht das Bearbeiten von XML-Dokumenten, ohne XML-Code Eingaben vorzunehmen.

Im Rahmen dieser Arbeit wird die Einsatzfähigkeit von WYSIWYG-fähigen Editoren für die Erstellung von XML-Templates in der Sächsischen Aufbaubank -Förderbank- untersucht. Hierzu wird eine Evaluation durchgeführt. Anschließend erfolgt die Erstellung eines Templates mittels eines ausgewählten Editors. Die Verwendung des erstellten XML-Templates mit dem ReportingTool JasperReports ist dabei maßgebend.

Anhand der gewonnenen Erkenntnisse, die aus dem praktischen Test des Editors hervorgehen, werden Lösungsvarianten zur zukünftigen Nutzung der WYSIWYG-fähigen Editoren aufgezeigt.

## Abkürzungsverzeichnis

SAB	Sächsische Aufbaubank -Förderbank-
XML	Extensible Markup Language
IDE	Integrierte Entwicklungsumgebung
WYSIWYG	What You See Is What You Get
EEV	Einheitliche Entscheidungsvorlage für Darlehen
DTD	Dokumenttyp Definition
XSD	XML Schema Definition
CSS	Cascading Style Sheets
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations
ISO	International Organization for Standardization
KWG	Kreditwesengesetz
TEI	Text Encoding Initiative
FOP	Formatting Objects Processor
PDF	Portable Document Format

## Abbildungsverzeichnis

Abbildung 1: Klassifizierung von XML-Dokumenten .....	8
Abbildung 2: Template- und Model-driven Mapping .....	10
Abbildung 3: Prinzip der Dokumentenerstellung mit JasperReports und ProSAB.....	12
Abbildung 4: ProSAB-Bearbeitungsansicht.....	14
Abbildung 5: Sektionen eines XML-Templates für JasperReports.....	16
Abbildung 6: Textlayout-Ansicht im XML-Editor XMLSpy von Altova .....	19
Abbildung 7: Gridlayout-Ansicht im Editor XMLSpy von Altova .....	20
Abbildung 8: WYSIWYG-Layout-Ansicht in Oxygen von SyncRO Soft.....	21
Abbildung 9: Kontextsensitive Eingabehilfe in Oxygen.....	41
Abbildung 10: XML-Code-Darstellung.....	42
Abbildung 11: Ausschnitt WYSIWYG- Layout-Ansicht in XMLSpy 2009.....	45
Abbildung 12: Ausschnitt WYSIWYG- Layout-Ansicht in Oxygen .....	45
Abbildung 13: Ausschnitt WYSIWYG- Layout-Ansicht in Serna.....	45
Abbildung 14: Entwurf einer Eingabemaske mit SPS-Stylesheet .....	51
Abbildung 15: Ausschnitt aus der Vorlage der EEV .....	56
Abbildung 16: Auswahlliste in ProSAB-Bearbeitungsansicht .....	59
Abbildung 17: Einfügen eines Parameters in WYSIWYG-Layout-Ansicht von Oxygen .....	59
Abbildung 18: Auswahllisten für die Eingabe von Attributwerten .....	59
Abbildung 19: Anlegen eines Freitextfeldes mit Vorbelegung .....	60
Abbildung 20: Freitextfeld in ProSAB-Bearbeitungsansicht .....	61
Abbildung 21: Erstellung der Entscheidungsvorlage in der ProSAB- Bearbeitungsansicht .....	62
Abbildung 22: unvollständige WYSIWYG-Layout-Ansicht in Oxygen.....	64

## Tabellenverzeichnis

Tabelle 1: Kriterien zum Produkt und Anbieter.....	28
Tabelle 2: Kriterien zu den allgemeinen Systemeigenschaften .....	28
Tabelle 3: Kriterien zur Benutzeroberfläche.....	29
Tabelle 4: Kriterien zur Dateneingabe und -bearbeitung in den Editoren .....	30
Tabelle 5: Kriterien zu den Prüfroutinen .....	30
Tabelle 6: Kriterien zur Softwarequalität.....	31
Tabelle 7: Kriterien zur Softwareergonomie .....	31
Tabelle 8: Bewertung der Anbieter und Systemeigenschaften für die ausgewählten Editoren.....	35
Tabelle 9: Bewertung der Anbieter und Systemeigenschaften für die aussortierten Editoren.....	37
Tabelle 10: Bewertung der Benutzeroberfläche .....	40
Tabelle 11: Bewertung der Dateneingabe.....	42
Tabelle 12: Bewertung der Prüfroutinen .....	43
Tabelle 13: Bewertung der Softwarequalität .....	44
Tabelle 14: Bewertung der Softwareergonomie .....	47
Tabelle 15: Gesamtbewertung der ausgewählten Editoren.....	50

# 1 Einleitung

Die Sächsische Aufbaubank -Förderbank- (SAB) ist das zentrale Förderinstitut des Freistaates Sachsen. In dessen Auftrag vergibt das Institut Fördermittel und zinsgünstige Kredite an Kommunen, Unternehmen und private Haushalte.

Auf der Basis von XML-Vorlagen erfolgt die Erstellung von Schriftstücken mit der hauseigenen Java-Applikation ProSAB, sie dient der Bearbeitung von Förderanträgen. Anhand der erstellten Dokumente erfolgt ein Teil des Schriftverkehrs zwischen der SAB und dem Antragsteller.

## 1.1 Motivation

Die Erstellung von XML-Vorlagen erfolgt in der SAB mit einem Editor. Dieser verfügt über eine Textlayout-Ansicht, die der manuellen Eingabe von XML-Code dient. Die Erstellung von Vorlagen innerhalb der Ansicht ist zeitaufwändig und setzt zwingend fachliche XML-Kenntnisse des Anwenders voraus.

Zur vereinfachten Erstellung von XML-Dokumenten in der Informationsverarbeitung wurden WYSIWYG (What You See Is What You Get)-Editoren entwickelt. In den vergangenen Jahren wurden zahlreiche Editoren um WYSIWYG-Layout-Ansichten ergänzt. Ein großer Vorteil von WYSIWYG-fähigen Editoren ist die Eigenschaft, dass dem Nutzer während der Erstellung des Dokuments bereits der visuelle Zustand der Daten auf dem Ausgabemedium vermittelt wird. Innerhalb der Ansicht hat der Nutzer die Möglichkeit beliebigen Text, Grafiken und Tabellen einzufügen. Während der Erstellung von XML-Dokumenten sind keine Code-Eingaben erforderlich.

Daher wäre es wünschenswert, wenn die Erstellung von XML-Vorlagen (Template) innerhalb der SAB künftig durch Editoren mit WYSIWYG-Layout-Ansicht erfolgen würde. Die Verwendung der neuen Ansicht würde zu einem verringerten Zeit- und Arbeitsaufwand der Erstellung des Templates führen. Weiterhin sind aufgrund der Dokumentvorschau weniger Tests und Kontrollen hinsichtlich des Layouts der erstellten Vorlage durchzuführen.

Die Editoren sollten allerdings den Anforderungen des Hauses, der eingesetzten Software ProSAB und dem verwendeten Reporting-Tool JasperReports entsprechen.

## **1.2 Voraussetzungen**

Der vorliegende Abschnitt stellt die gegebenen Voraussetzungen und Rahmenbedingungen zur Durchführung der Untersuchung sowie der prototypischen Erstellung des XML-Templates „Einheitliche Entscheidungsvorlage für Darlehen“ (EEV) dar.

Die Durchführung einer Evaluation von XML-Editoren geschieht unter den gegebenen Bedingungen der SAB. Für nachfolgende praktische Einsatztests der Editoren wird zur prototypischen Erstellung des XML-Templates EEV eine Vorlage bereitgestellt, die Aufbau, Formatierung und Inhalt des Dokuments festlegt. Für die Umsetzung der EEV stehen vorhandene ProSAB-Templates (Bausteine) zur Verfügung, die in das zu erstellende XML-Template integriert werden können. Die EEV stellt ein umfangreiches Dokument zur Herbeiführung einer Entscheidung über die Annahme bzw. Ablehnung eines Kreditantrags dar. Das Dokument enthält zahlreiche ProSAB-spezifische Elemente sowie verschiedene Bausteine. Aufgrund des vielfältigen Dokumentinhalts werden während des Editorentests die gesamten Funktionsansprüche der SAB abgedeckt.

Der zu erstellende Prototyp wird gegen eine vorgegebene Dokumenttyp Definition (DTD) validiert. In dieser sind zu verwendende Elemente und Attribute des XML-Templates definiert. Zum anschließenden Test des Prototyps wird die Applikation ProSAB zur Verfügung gestellt.

## **1.3 Aufgabenstellung und Ziele**

Im Rahmen der Bachelorarbeit sind WYSIWYG-fähige Editoren, die zur Erstellung von XML-Templates für ProSAB geeignet sind, hinsichtlich ihrer Beschaffenheit und Einsatzmöglichkeit zu untersuchen. Ziel der Untersuchung ist eine Vereinfachung der Templateerstellung, sodass auch Nutzer, die im Umgang mit XML nicht versiert sind, die Möglichkeit haben XML-Dokumente zu erstellen. Die Evaluation der Editoren erfolgt anhand von Kriterien, die auf den Anforderungen von ProSAB, JasperReports und dem Einsatzumfeld innerhalb der SAB beruhen. Ausgewählte Editoren sind anschließend unter den gegebenen Bedingungen der SAB zu testen. Dabei ist das hausinterne ProSAB-Template EEV prototypisch zu erstellen.

Für die Erstellung von Templates sind ausschließlich WYSIWYG-fähige Editoren zu untersuchen. Anderweitige Formen der Erstellung von XML-Vorlagen, wie Editoren, die lediglich über eine Textlayout-Ansicht verfügen oder Dokumente in das XML-Format konvertieren können, finden keine Berücksichtigung.

---

Die Anwendung der Textlayout-Ansicht zur Templateerstellung wird bereits praktiziert. Daher ist das Funktionsangebot weiterer Bearbeitungsansichten von Interesse.

## **1.4 Aufbau der Arbeit**

Im Folgenden werden die einzelnen Abschnitte der Bachelorarbeit dargestellt.

Das Kapitel 2 gibt grundlegende Informationen zum Themengebiet der vorliegenden Arbeit. Außerdem werden die Metasprache XML, das Reporting-Tool JasperReports und der Aufbau von WYSIWYG-fähigen Editoren erläutert.

Im Kapitel 3 wird das Vorgehen zur Auswahl der Editoren aufgezeigt. Zu Beginn der Evaluation werden Bewertungskriterien aufgestellt. Auf Grundlage der ermittelten Kriterien wird anschließend eine Erhebung von Anbieter- und Produktinformationen vorgenommen. Danach folgen die Auswertung der erfassten Informationen und eine Produktauswahl.

Das Kapitel 4 gibt detaillierte Informationen zur Erstellung des Prototyps EEV mittels eines ausgewählten WYSIWYG-fähigen Editors. Anschließend erfolgt die Auswertung der gewonnen Erkenntnisse.

Im Kapitel 5 erfolgt die Vorstellung der Ergebnisse, die aus der Arbeit hervorgehen. Weiterhin werden die praktischen Einsatzmöglichkeiten von XML-Editoren zur Templateerstellung in der SAB aufgezeigt.

## 2 Theoretische Grundlagen

Zum besseren Verständnis der vorliegenden Arbeit werden nachfolgend relevante Begriffe und Technologien erläutert. Dabei wird näher auf den XML-Standard, ProSAB-Templates, JasperReports und WYISWYG-fähige Editoren eingegangen.

### 2.1 XML-Allgemein

Die Extensible Markup Language (XML) basiert auf der Standard Generalized Markup Language (SGML), die zu den Metasprachen zählt und bereits in den 70er Jahren bei IBM erfunden wurde<sup>1</sup>. Mittels der Metasprache SGML wurde im Jahre 1989 die Auszeichnungssprache Hypertext Markup Language (HTML) festgelegt.

Im Jahre 1998 wurde der XML 1.0 Standard durch das World Wide Web Consortium (W3C) veröffentlicht. Das W3C ist ein internationales Konsortium, das sich aus zahlreichen Mitgliedsorganisationen zusammensetzt. Ein bestehendes Team entwickelt gemeinsam mit der Öffentlichkeit Web-Standards.

Auszeichnungssprachen (Markup Language) ermöglichen die Kennzeichnung von Textbestandteilen mittels Tags (Markups), in diesem Zusammenhang werden dem Textelement zugleich Eigenschaften zugewiesen. Der zu kennzeichnende Teil des Textes wird durch Start- und End-Tags umschlossen.

XML ermöglicht die Definition von Auszeichnungssprachen mit eigenen Tags. Hierfür werden selbstdefinierte Elemente und Attribute vereinbart. Markup-Sprachen, die mit XML definiert sind, werden als XML-Anwendungen bezeichnet<sup>2</sup>.

XML-Dokumente basieren auf einer Struktur, sie kann in Form einer Dokumenttyp-Definition (DTD) oder einer XML-Schema-Definition (XSD) vorgegeben werden. Die Definitionen beschreiben den Aufbau eines XML-Dokuments, weiterhin werden die erlaubten Elemente und Kindelemente sowie deren Attribute festgelegt.

In XML-Dokumenten werden Daten als Zeichenketten eingefügt, die von Tags umschlossen sind. Anhand der Tags erfolgt eine Beschreibung der Daten, wodurch der XML-Code für einen Außenstehenden leichter zugänglich ist. „Eine XML-Basiseinheit aus Daten und Markup wird Element genannt.“<sup>3</sup> Ein Element kann wie folgt dargestellt werden:

---

<sup>1</sup> vgl. HAROLD, E. R./MEANS, W. S. (2005), S. 9

<sup>2</sup> vgl. PARTL, H. (2007)

<sup>3</sup> HAROLD, E. R./MEANS, W. S. (2005), S. 3

---

```
<startTag>Mustermann</endTag>
```

Die Verwendung von Attributen ermöglicht eine nähere Beschreibung der Elemente. Ein Attribut besteht jeweils aus einem Namen und einem Wert, der Attributwert wird in Anführungszeichen angegeben und durch das Gleichheitszeichen dem Namen zugewiesen. Die Angabe des Attributs erfolgt im Start-Tag eines Elements.

```
<startTag geboren="12-03-1980">Mustermann</endTag>
```

Bei der Erstellung von XML-Dokumenten sind syntaktische Regeln<sup>4</sup> zu beachten. Hierzu zählen die Verwendung von Unicode-Zeichen, Groß- und Kleinschreibung, Prolog<sup>5</sup> und der einmalige Aufruf des Wurzelements<sup>6</sup>. Weiterhin sind Elemente sauber ineinander zu verschachteln, das heißt es müssen geöffnete Tags, die verschachtelt sind in umgekehrter Reihenfolge wieder geschlossen werden. Außerdem bestehen verwendete Elemente jeweils aus einem Start- und End-Tag, andernfalls sind sie als leere Elemente wie folgt anzugeben:

```
<leererTag/>
```

Sofern XML-Dokumente diese syntaktischen Regeln einhalten werden sie als wohlgeformt bezeichnet.

Weiterhin müssen XML-Dokumente der vorgegebenen DTD bzw. XSD entsprechen, indem sie die definierte Struktur einhalten. Wenn diese Bedingung erfüllt ist wird das Dokument als gültig bzw. valid bezeichnet. Die Gültigkeit eines XML-Dokuments setzt die Wohlgeformtheit voraus.<sup>7</sup>

Die Erstellung von DTDs erfolgt mittels einer formalen Syntax. Sie definiert die zulässigen Elemente sowie deren Verwendung im XML-Dokument. Außerdem ist es möglich jedem Element eigene Attribute zuzuweisen.

---

<sup>4</sup> Die XML-Spezifikation wird durch das W3C herausgegeben.

<sup>5</sup> Die Angabe der XML-Version erfolgt am Dokumentanfang.

<sup>6</sup> Das Wurzelement ist das erste Element, das nachdem Prolog angegeben wird, es wird auch Root-Element genannt.

<sup>7</sup> vgl. HAROLD, E. R./MEANS, W. S. (2005), S. 33

---

Im Gegensatz zu einer DTD beschreibt die XML Schema Definition die Struktur von XML-Dokumenten mittels XML. Zusätzlich wird eine Vielzahl von Datentypen unterstützt z.B. Boolean, String und Float.

Für die Verwendung von Inhalten aus XML-Dokumenten in einer Anwendung wird ein Parser benötigt. Ein Parser ist ein Programm, das z.B. XML-Dokumente liest und die Aufteilung des Dokuments in Elemente, Attribute und andere Bestandteile vornimmt. Den ermittelten Inhalt des XML-Dokuments übergibt der XML-Parser anschließend an die Anwendung. Diese kann den übermittelten Inhalt weiter verarbeiten.

Zum Parsen von Dokumenten stehen mehrere XML-Parser zur Verfügung. Sie unterscheiden sich voneinander in folgenden Eigenschaften: Validierung und Schnittstelle für den Zugriff auf das XML-Dokument. Ein validierender XML-Parser überprüft das zu lesende Dokument auf Wohlgeformtheit, weiterhin werden die enthaltenen Elemente und Attribute mit der DTD bzw. XSD verglichen. Sobald der Parser eine Verletzung der XML-Regeln zur Wohlgeformtheit bzw. einen Verstoß gegen die DTD oder XSD feststellt, bricht er das Parsen des XML-Dokuments ab. Anschließend kann eine Fehlermeldung durch die Anwendung ausgegeben werden. Weiterhin unterscheiden sich XML-Parser in der verwendeten Schnittstelle, die den Zugriff auf das XML-Dokument ermöglicht. Dabei sind als namhafte Vertreter The Simple API for XML (SAX) und Document Object Model (DOM) zu benennen.

XML-Dokumente bieten den Vorteil, dass sie beliebigen Text speichern können. Diese Eigenschaft ermöglicht das Lesen des Dokuments ohne Zuhilfenahme von Anwendungen. Außerdem können die Anwender eigene DTDs und XSD-Schemas erstellen, die den Dokumenten eine Struktur vorgeben. Für die Verarbeitung des Dokumentinhalts in Anwendungen stehen zahlreiche kostenlose Parser zur Verfügung, die in eigene Applikationen eingebunden werden können. Die genannten Vorzüge führen dazu, dass XML ein weitverbreiteter Standard ist.

### **2.1.1 XML in der Praxis**

Prinzipiell sind Unternehmen bestrebt ihre betriebsinternen Prozesse zu optimieren. Aus diesem Grund werden z.B. in Industrieunternehmen oftmals Bestellungen automatisch ausgelöst, sobald der Materialbestand den Mindestbestand unterschreitet. Jedoch können im anschließenden Datenaustausch Probleme auftreten und die Bestellungen beim Lieferanten unmöglich machen. Die Störungen im Datenaustausch werden unter anderem durch unterschiedliche Warenwirtschaftssysteme von Lieferanten und den verarbei-

tenden Unternehmen hervorgerufen<sup>8</sup>. Dieses schwerwiegende Kompatibilitätsproblem betrifft zahlreiche Unternehmen und Institutionen auf der Welt. Um dem Problem zu entgehen wird meistens das plattformneutrale Datenformat XML eingesetzt<sup>9</sup>. Das XML-Datenformat stellt hierbei nur eine Möglichkeit dar, es bestehen weitere Standarddatenformate wie „Electronic Data Interchange For Administration, Commerce and Transport“ (EDIFACT). Die am Datenaustausch beteiligten Einrichtungen nutzen eine gemeinsam entworfene Struktur z.B. eine DTD. Sie wird in die jeweilige Anwendungssoftware integriert und ermöglicht dadurch das korrekte Auslesen der Daten aus dem XML-Dokument.

Neben dem Austauschen von Daten zwischen Einrichtungen eignet sich XML auch hervorragend als Speicherformat für Daten. Ausschlaggebend dafür sind mitunter die einfache Syntax der XML-Dokumente sowie die Möglichkeit, mit Hilfe von geschachtelten Elementen, komplexe Datenstrukturen abzubilden.

Eine weitere Einsatzmöglichkeit von XML ist die Erstellung von Webseiten. Dabei wird der dargestellte Text der Webpräsentation in einem XML-Dokument gespeichert, die Formatierung des auszugebenden Textes erfolgt mit Cascading Style Sheets (CSS) bzw. Extensible Stylesheet Language Transformations (XSLT). Durch Zuhilfenahme eines Parsers können aus bestehenden XML-Dokumenten neue Dokumente erstellt werden (z.B. in HTML).

XML-Dokumente können unter anderem auch in Form von Templates eingesetzt werden. Sie fungieren als Schablonen, auf deren Basis neue Dokumente in einem anderen Format generiert werden können. Im XML-Template werden dabei unter Verwendung einer DTD bzw. XSD die Formatierung sowie der Inhalt des Dokuments vorgegeben. Das zu erstellende XML-Dokument wird durch eine Anwendung, die das XML-Template und einen Parser z.B. den Formatting Objects Processor (FOP) implementiert, erzeugt. Es besteht weiterhin die Möglichkeit Dokumente in anderen Ausgabeformaten, wie das Portable Document Format (PDF) zu erstellen. Die Erstellung von XML-Templates und die Erzeugung von Dokumenten im XML- und PDF-Format, unter der Verwendung des Reporting-Tools JasperReports, werden in den nachfolgenden Kapiteln näher dargestellt.

---

<sup>8</sup> vgl. TIEMEYER, E./FEIL, T. (2007), S. 568

<sup>9</sup> vgl. HAROLD, E. R./MEANS, W. S. (2005), S. 266

## 2.1.2 Klassifizierung von XML-Dokumenten

XML-Dokumente lassen sich anhand des Inhalts und der Struktur in unterschiedliche Dokumenttypen einteilen. Hierbei werden die dokumentzentrierte, datenzentrierte und semistrukturierte Klasse unterschieden (siehe Abbildung 1).

dokumentzentriert	datenzentriert	semistrukturiert
<pre> &lt;block&gt;   &lt;ueberschrift&gt;     XML   &lt;/ueberschrift&gt;   &lt;absatz&gt; XML basiert auf der Standard Generalized Markup Language (SGML), die zu den Metasprachen zählt.   &lt;/absatz&gt;   &lt;absatz&gt;     ...   &lt;/absatz&gt; &lt;/block&gt; </pre>	<pre> &lt;bestellung&gt;   &lt;name&gt;     Mustermann, Manfred   &lt;/name&gt;   &lt;artikel&gt;     &lt;titel&gt;       Einführung XML     &lt;/titel&gt;     &lt;isbn&gt;       123     &lt;/isbn&gt;   &lt;/artikel&gt; &lt;/bestellung&gt; </pre>	<pre> &lt;buch&gt;   &lt;autor&gt;     Mustermann, Manfred   &lt;/autor&gt;   &lt;titel&gt;     Einführung XML   &lt;/titel&gt;   &lt;isbn&gt;     123   &lt;/isbn&gt;   &lt;inhalt&gt; XML basiert auf der Standard Generalized Markup Language (SGML), die zu den Metasprachen zählt.   &lt;/inhalt&gt; &lt;/buch&gt; </pre>

Abbildung 1: Klassifizierung von XML-Dokumenten

Dokumentzentrierte Dokumente dienen der Darstellung von Informationen, die hauptsächlich aus zusammenhängenden Texten bestehen. Daher werden sie vor allem für Bücher, Zeitschriftenbeiträge, Gebrauchsanweisungen und Verträge verwendet. Die Struktur dieser XML-Dokumente ist stets unterschiedlich, obwohl innerhalb des Dokuments z.B. Überschriften-, Absatz- und Block-Elemente verwendet werden<sup>10</sup>. Allerdings ist die Reihenfolge der Elemente von hoher Bedeutung, da eine Änderung der Anordnung zu einem Informationsverlust führt.

Datenzentrierte Dokumente sind hingegen stark strukturiert und stellen vor allem Daten aus relationalen Datenbanken dar<sup>11</sup>. Außerdem sind wichtige Informationen meistens in Blattelementen und Attributen enthalten. Aus diesem Grund eignen sich datenzentrierte Dokumente für die Abbildung von Fahrplänen, Telefonbüchern, wissenschaftlichen Arbeiten und Bestellungen. Die Erstellung der Dokumente erfolgt in der Regel

<sup>10</sup> vgl. SCHÖNING, H. (2003), S. 65

<sup>11</sup> vgl. Ebenda, S. 64

---

maschinell. Die Daten werden hierbei aus relationalen Datenbanken importiert bzw. können aus dem Dokument in Datenbanken exportiert werden.

Die Zuordnung von XML-Dokumenten zu einer der vorgestellten Klassen ist schwierig, da Dokumente meistens nicht allen Merkmalen einer Klasse entsprechen. Eine Mischform die Merkmale der dokumentenzentrierten und datenzentrierten Dokumente enthält wird als semistrukturiert bezeichnet. Semistrukturierte Dokumente sind dadurch gekennzeichnet, dass sie strukturierte und datentypbehaftete Informationen enthalten, die mit textuellen Elementen gemischt sind<sup>12</sup>. Demzufolge enthalten semistrukturierte Dokumente neben Text auch Daten aus relationalen Datenbanken oder Inhalt aus anderen XML-Dokumenten.

### **2.1.3 Abbildung von Daten zwischen XML-Dokumenten und relationalen Datenbanken**

Es besteht die Möglichkeit, ein komplettes XML-Dokument auf eine relationale Datenbank<sup>13</sup> abzubilden. Dieses Verfahren wird Mapping genannt. Für das Erstellen und Speichern von Daten in einem XML-Dokument bzw. einer relationalen Datenbank muss die Struktur des XML-Dokuments bekannt sein, die in einer DTD oder einem XML-Schema definiert ist.

Daten aus dokumentenzentrierten XML-Dokumenten können als Binary Large Object (BLOB) abgelegt werden. Dabei wird der ganze Inhalt des XML-Dokuments in einem Datensatz gespeichert.

Unter den datenzentrierten XML-Dokumenten werden die Ansätze des Template-driven und Model-driven Mappings verfolgt<sup>14</sup>.

Das Template-driven Mapping ermöglicht lediglich den Transfer von Daten aus relationalen Datenbanken in XML-Dokumente. Hierbei besteht keine Abbildung zwischen der XML-Dokument- und der Datenbankstruktur. Die Erstellung des XML-Dokuments geschieht mittels XML-Templates, die z.B. eingebettete Select-Statements enthalten<sup>15</sup>. Diese werden bei der Erstellung des XML-Dokuments vom Server verarbeitet.

Umfangreichere Möglichkeiten bietet das Model-driven Mapping. Es ermöglicht die Abbildung einer XML-Struktur auf eine Datenbankstruktur. Das Table-Model und das

---

<sup>12</sup> vgl. SCHÖNING, H. (2003), S. 67

<sup>13</sup> Daten werden in Form von Tabellen gespeichert, jede Tabellenzeile erhält zur Identifizierung einen eindeutigen Schlüssel.

<sup>14</sup> vgl. BOURRET, R. (2005)

<sup>15</sup> vgl. Ebenda

datenspezifische Object-Model sind in diesem Zusammenhang die bekanntesten Vertreter, ihre Struktur ist in Abbildung 2 dargestellt.

Template driven Mapping	Model-driven Mapping	Model-driven Mapping
<pre data-bbox="284 488 542 795"> &lt;liste&gt;   &lt;text&gt;     Teilnehmerliste:   &lt;/text&gt;   &lt;selectStmt&gt;     SELECT Name,     Vorname, Alter     FROM Personen   &lt;/selectStmt&gt; &lt;/liste&gt; </pre>	<pre data-bbox="675 488 933 1057"> &lt;liste&gt;   &lt;text&gt;     Teilnehmerliste:   &lt;/text&gt;   &lt;tabelle&gt;     &lt;zeile&gt;       &lt;name&gt;         Muster       &lt;/name&gt;       &lt;vorname&gt;         Klaus       &lt;/vorname&gt;       &lt;alter&gt;         45       &lt;/alter&gt;     &lt;/zeile&gt;     ...   &lt;/tabelle&gt; &lt;/liste&gt; </pre>	<pre data-bbox="1066 488 1364 884"> &lt;liste&gt;   &lt;text&gt;     Teilnehmerliste:   &lt;/text&gt;   &lt;person nr="1"&gt;     &lt;name/&gt;     &lt;anschrift     sitz="hauptwohnsitz"&gt;       &lt;strasse/&gt;       &lt;ort/&gt;     &lt;/anschrift&gt;   &lt;/person&gt; &lt;/liste&gt; </pre>

Abbildung 2: Template- und Model-driven Mapping

Das Table-Model ermöglicht eine flache Abbildung zwischen XML-Dokumenten und Tabellen relationaler Datenbanken. Jedoch können durch das Table-Model nur XML-Dokumente mit einer Tiefe von höchstens 3 geschachtelten Elementen abgebildet werden, daher ist es nur für relationale Datenbanken verwendbar.

Beim datenspezifischen Object-Model werden die Daten modelliert. Sie werden als Objekte dargestellt und können als hierarchische Baumstruktur abgebildet werden. Das entstandene Modell kann anschließend auf objektorientierte oder hierarchische Datenbanken abgebildet werden<sup>16</sup>.

#### 2.1.4 Erstellung von XML-Templates für ProSAB

In der SAB erfolgt der Schriftverkehr mit den Kunden unter den Aspekten des Corporate Designs. Für das einheitliche Erscheinungsbild der Institution liegt allen Dokumenten ein vorgegebenes Layout zugrunde. Hierzu werden für Schriftstücke in ProSAB XML-Templates bereitgestellt. Templates stellen Schablonen für Dokumente dar. Mittels

<sup>16</sup> vgl. KLEIN, O. (2001), S. 14

---

XML wird die genaue Struktur des zu erstellenden Schriftstücks im Template definiert. Das in ProSAB erzeugte Schriftstück wird anschließend im XML- und PDF-Format gespeichert.

XML-Templates beinhalten vor allem Text und eingebettete Verweise. Die Referenzierung auf das Java-Objektmodell erfolgt mit ProSAB-Keys und -Listen. Die genannten Elemente sind mit eingebetteten Selects vergleichbar, allerdings greifen sie auf das Java-Objektmodell von ProSAB zu. Das Objektmodell übergibt die Daten einer relationalen Datenbank an das Template.

Weiterhin können im Template Textdaten aus anderen Bausteinen (XML-Templates) enthalten sein, die durch Subreports eingebunden sind.

Aufgrund dessen, dass jedes XML-Template einem spezifischen Schriftstück als Vorlage dient, weicht die Struktur der Templates voneinander ab. Die Vielfalt der Bestandteile, z.B. Text oder Tabellen mit Daten aus relationalen Datenbanken, innerhalb der ProSAB-Templates, lässt schlussfolgern, dass sie als semistrukturiert zu klassifizieren sind.

Die Entwicklung von XML-Templates erfolgt mit dem XML-Editor XMLSpy des Herstellers Altova. Unter Verwendung der Textlayout-Ansicht geschieht die Eingabe des XML-Codes. Das zu erstellende XML-Template wird gegen eine DTD validiert. Diese basiert auf der Vorgabe von JasperReports<sup>17</sup>, wurde allerdings um ProSAB-spezifische Elemente wie ProSAB-Keys und -Listen erweitert.

## **2.2 Zusammenwirken von XML-Templates und JasperReports**

Die Bearbeitung von Förderanträgen, inklusive dem Schriftverkehr, geschieht in der SAB mittels der Client-Server-Anwendung ProSAB. Basierend auf dem Client-Server-Modell besitzt jeder Rechner die Anwendung als Client. Über das Netzwerk greift die Anwendung auf den Applikationsserver zu. Der ProSAB-Client entspricht hierbei einem Thin-Client<sup>18</sup>.

Innerhalb der ProSAB-Bearbeitungsansicht werden die enthaltenen Daten eines Templates sowie die abgefragten Informationen des Datenbankservers dargestellt.

---

<sup>17</sup> Ein Reporting-Tool, mit dem dynamisch generierte, grafische Erzeugnisse auf Monitor und Drucker sowie in unterschiedlichen Datenformaten ausgegeben werden.

<sup>18</sup> Bei einem Thin-Client erfolgt die Verarbeitung von Daten durch einen Server, auf dem Einzelplatzrechner befindet sich lediglich ein Client mit Swing-Oberfläche. Die Client-Anwendung basiert auf dem Ultra Light Client (ULC) des Software-Unternehmens Canoo.

## 2.2.1 Erzeugung von Dokumenten mittels ProSAB und JasperReports

In dieser Arbeit wurde bereits erwähnt, dass die Erstellung eines Dokuments im PDF- und XML-Format, durch ProSAB auf Grundlage von XML-Templates erfolgt. Im Folgenden wird dieser Prozess veranschaulicht (siehe Abbildung 3).

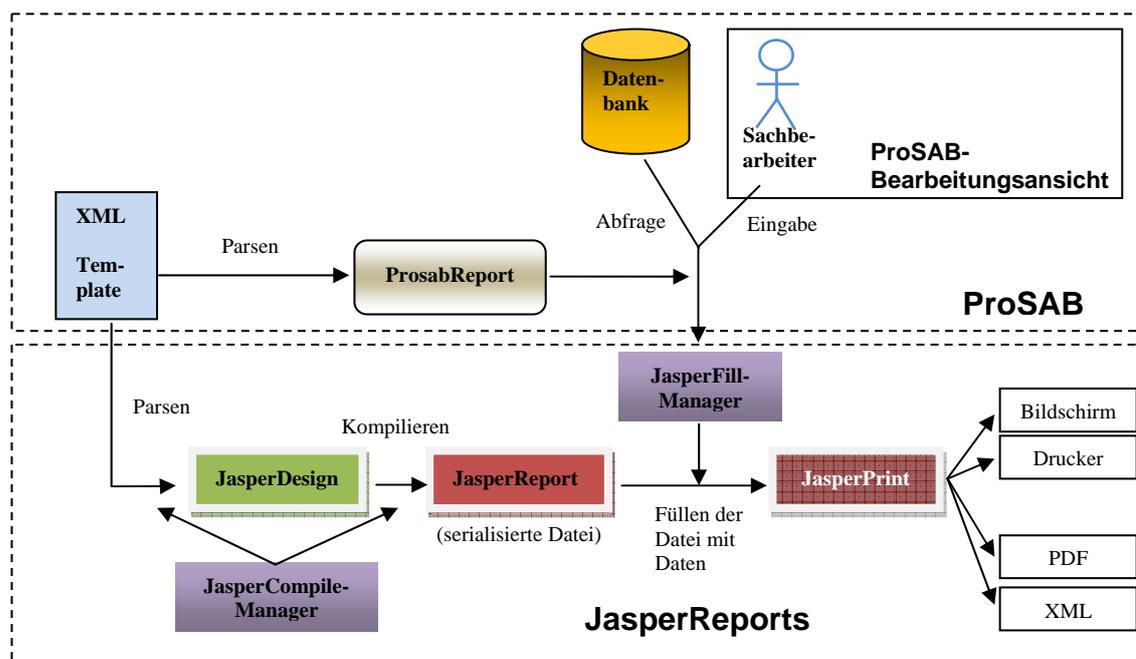


Abbildung 3: Prinzip der Dokumentenerstellung mit JasperReports und ProSAB<sup>19</sup>

Im Rahmen der Antragsbearbeitung enthält ProSAB Arbeitsschritte, die der Erstellung von Dokumenten dienen. Bevor allerdings dem Sachbearbeiter die Bearbeitungsansicht bereitgestellt wird, muss das zu einem Arbeitsschritt zugeordnete XML-Template geparkt werden. Dies erfolgt durch das Reporting-Tool JasperReports unter der Verwendung des Parsers SAX 2.0<sup>20</sup>. JasperReports enthält den JasperCompileManager, der das XML-Template sowie die enthaltenen Bausteine parst. Dabei werden die XML-Templates auf Wohlgeformtheit überprüft und gegen die zugewiesene ProSAB-DTD validiert. Anschließend erzeugt JasperReports ein JasperDesign, welches alle Elemente mit Attributen in Form von Objekten enthält. Darauf folgt die Kompilierung, innerhalb derer der JasperCompileManager das JasperDesign verarbeitet und eine serialisierte Datei, den JasperReport erstellt. Während des Vorgangs werden Elemente wie ProSAB-Keys und -Listen ignoriert, da sie JasperReports unbekannt sind. Daher ist in

<sup>19</sup> In Anlehnung an DANCIU, T. (2002), S. 5, Abb. 1

<sup>20</sup> vgl. DANCIU, T. (2002), S. 10

---

Abbildung 3 ein weiterer Vorgang dargestellt, in welchem das XML-Template durch ProSAB geparkt wird. Dabei erfolgt jedoch nur ein Mapping von ProSAB-Elementen, die der ursprünglichen JasperReports-DTD hinzugefügt wurden.

Auf Grundlage des erstellten ProsabReports ermittelt ProSAB mittels Objektzugriffen und Eingabefeldern der ProSAB-Bearbeitungsansicht die benötigten Daten. Sie werden nachfolgend an den JasperFillManager überreicht. Der JasperFillManager füllt den JasperReport, der lediglich serialisierte Daten von JasperReport-Elementen des XML-Templates enthält. Die fehlenden Daten werden nun durch den JasperFillManager ergänzt. Im Anschluss an den Füllprozess wird das serialisierte Objekt JasperPrint gespeichert. Auf dessen Grundlage erzeugt JasperReports Ausgaben am Monitor, Drucker und unter anderem Dokumente im XML- und PDF-Format.

## 2.2.2 Antragsbearbeitungs-Software ProSAB

Die Antragsbearbeitung erfolgt in ProSAB anhand eines vordefinierten Workflows, der den Sachbearbeiter durch die einzelnen Arbeitsschritte führt. Unter den Arbeitsschritten verbirgt sich mitunter das Erstellen von Bescheiden und diversen Anschreiben. Für die Erstellung von Schriftstücken ist es unter anderem notwendig, dass der Sachbearbeiter ergänzende Eingaben vornimmt. Hierfür wird eine Bearbeitungsansicht (siehe Abbildung 4) bereitgestellt.

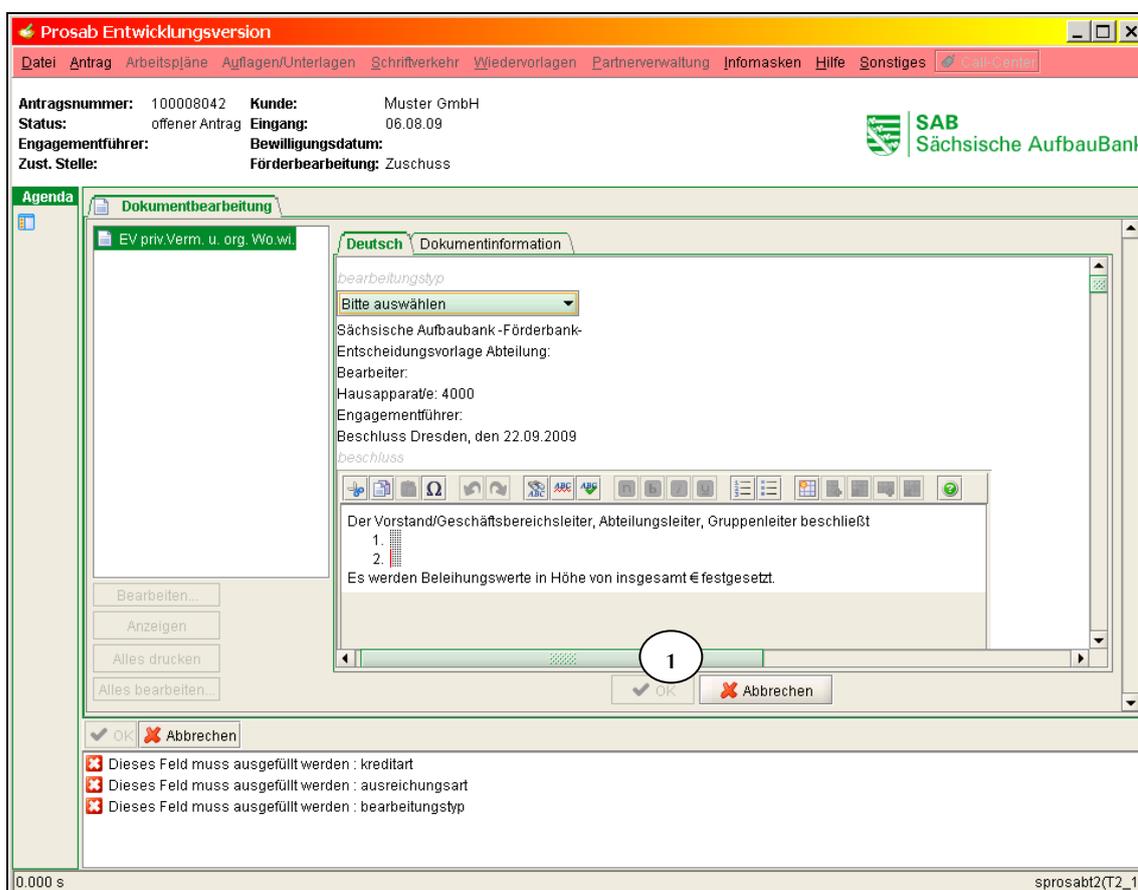


Abbildung 4: ProSAB-Bearbeitungsansicht

Im dargestellten Beispiel sind bereits Textpassagen des Schriftstücks enthalten. Weiterhin stehen dem Sachbearbeiter diverse Eingabefelder und Auswahllisten für Ergänzungen des Dokuments zur Verfügung. In Abbildung 4 ist ein Freitextfeld mit der Ziffer eins gekennzeichnet.

Die zum Teil vorbelegten Texte und Eingabefelder sowie deren Anordnung beruhen auf XML-Templates. Jedes Template gibt die Struktur eines Schriftstücks vor. Daher sind Anschrift, Textpassagen und Tabellen größtenteils im XML-Template enthalten bzw. es

---

wird für die Ausgabe von Daten auf das Java-Objektmodell von ProSAB referenziert. Das vervollständigte Schriftstück wird anschließend als PDF-Dokument dem Sachbearbeiter angezeigt.

### **2.2.3 Reporting-Tool JasperReports**

Reporting-Tools dienen der Erstellung, Speicherung und Verarbeitung von Berichten. Diese enthalten z.B. Daten zur Überprüfung der strategischen Unternehmensziele oder Statistiken.

Eines der bekanntesten Open Source Reporting-Tools für die Programmiersprache Java ist JasperReports. Es wurde zum ersten Mal im November 2001 durch Theodor Danciu veröffentlicht. Das Reporting-Tool stellt keine eigenständige Anwendung dar, sondern bedarf des Aufrufs der JasperReports-Klassenbibliotheken innerhalb der auszuführenden Java-Applikation<sup>21</sup>.

JasperReports ermöglicht die Erstellung von umfangreichen Berichten, die neben Text auch Grafiken und Diagramme enthalten. Die Gestaltung des Berichtlayouts ist dem Anwender überlassen, eine Beschränkung liegt lediglich in den zu verwendenden Elementen der DTD vor. Die Elemente und Attribute werden durch JasperReports als Objekte von Java-Klassen dargestellt. Eine Änderung der DTD bzw. Anpassung des Tools an eigene Bedürfnisse ist nicht vorgesehen. In der SAB wurde jedoch die JasperReports DTD erweitert, sodass ein zusätzlicher Parser für die hinzugefügten Elemente zu verwenden ist.

Es besteht die Möglichkeit, in Berichten Informationen aus unterschiedlichen Datenquellen zu integrieren. Sie können unter anderem aus Datenbanken und XML-Dokumenten abgerufen werden. Außerdem ist es möglich über Java-Schnittstellen (Datasources) Objekte auszuwerten. Diese Form der Datenbereitstellung wird in ProSAB-Templates überwiegend verwendet.

### **2.2.4 Struktur eines ProSAB-Templates**

ProSAB-Templates müssen dem vorgegebenen XML Standard (Wohlgeformtheit, Validierbarkeit) des W3C entsprechen. Die Reihenfolge und Hierarchie der Elemente im Template basieren auf der vorgegebenen ProSAB-DTD.

---

<sup>21</sup> vgl. HEFFELFINGER, D. R. (2006), S. 8

Die Templates entsprechen im Aufbau Reports (Berichten), die für JasperReports erstellt werden. Reports können unter anderem aus folgenden Sektionen bestehen: Titel (report title), Kopfzeile (page header), Dokumentinhalt (detail section) und Fußzeile (page footer)<sup>22</sup>. Außer den genannten Sektionen, bestehen weitere, die in XML-Templates für ProSAB angewandt und nachfolgend näher erläutert werden, (siehe Abbildung 5).

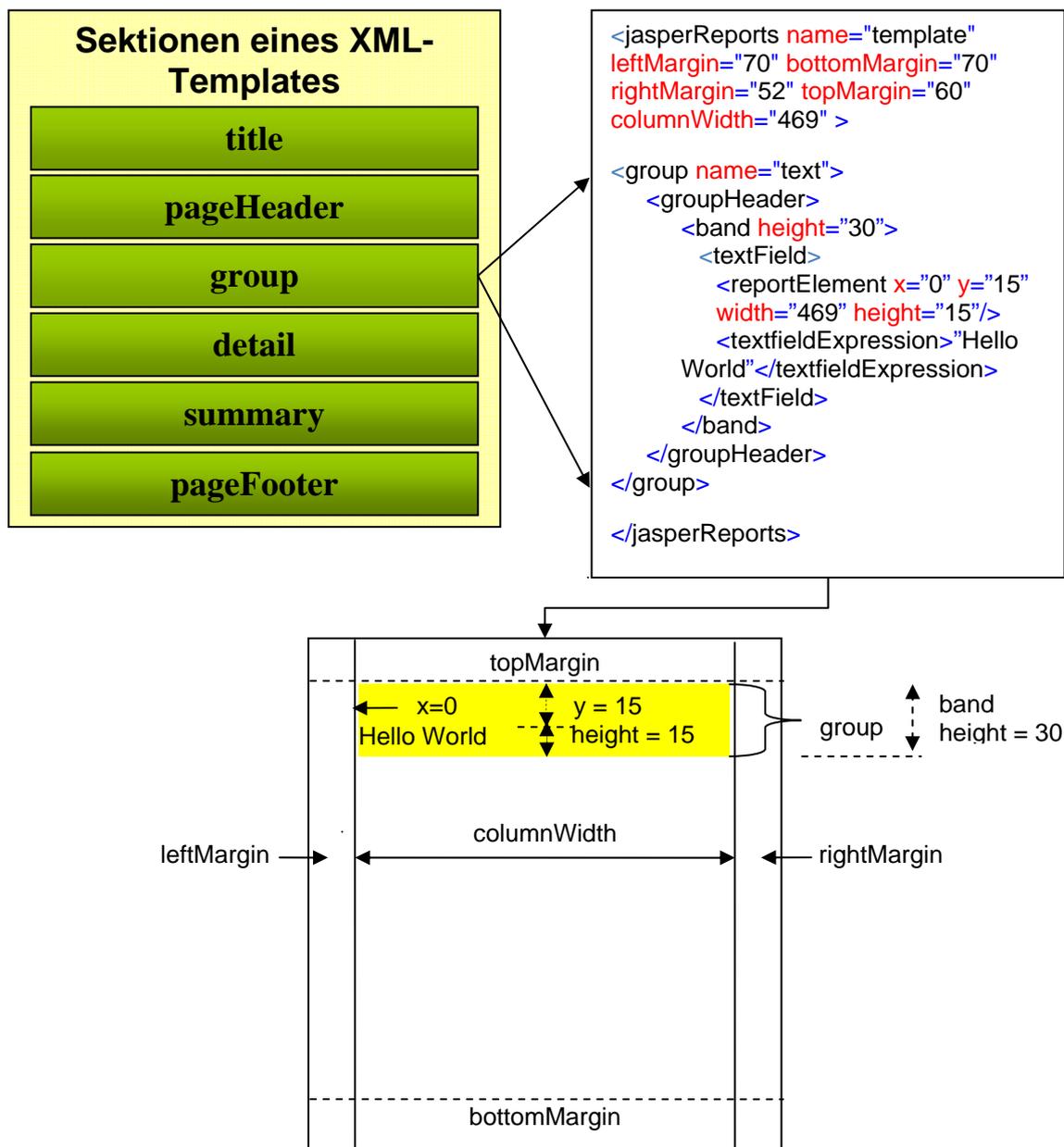


Abbildung 5: Sektionen eines XML-Templates für JasperReports

<sup>22</sup> vgl. HEFFELFINGER, D. R. (2006), S. 9

Jedes ProSAB-Template besitzt entsprechend dem XML-Standard einen Prolog, in dem die Angabe des Dokumenttyps enthalten ist.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jasperReport PUBLIC "-//JasperReports//DTD
Report Design//EN" "prosab_jasperreport.dtd">
```

Dem Prolog folgt das Wurzelement (Root-Element) des Templates, es lautet `<jasperReports>` und enthält zahlreiche Attribute, die der Definition des Template-Layouts dienen. Unter anderem werden Seitenhöhe, Seitenbreite und die Seitenränder des Schriftstücks vereinbart.

In der DTD ist die Reihenfolge zur Anordnung der Elemente in ProSAB-Templates angegeben. Dementsprechend müssen zu Beginn des XML-Codes im Template die verwendeten Elemente: `<filter>`, `<tableFilter>`, `<parameter>` und `<variable>` vereinbart werden. Außer der `<variable>` enthalten die genannten Elemente ProSAB-Keys, -Listen und viele weitere spezifische Elemente.

Es folgt die Angabe bzw. Definition des Inhalts, der letztendlich im Zieldokument erscheint. In der SAB wird dazu das `<group>` Element verwendet. Es ermöglicht eine logische und zusammenhängende Anordnung des Dokumentinhalts. Ferner kann jedes `<group>` Element einen `<groupHeader>`, `<groupFooter>` und `<groupExpression>` enthalten. Die genannten Elemente verfügen über das Kindelement `<band>`. Es fasst alle enthaltenen Elemente wie `<textField>`, `<subreport>` und grafische Elemente zusammen. Somit ist eine geschlossene Ausgabe der darin befindlichen Texte und Daten auf einer Dokumentseite möglich. Innerhalb eines `<band>` sind alle Elemente frei positionierbar. Dazu sind die Koordinaten im `<reportElement>` dementsprechend anzugeben. Ein bedeutendes Kindelement des `<band>` Elements ist das `<textField>`. Es dient der Ausgabe von Text und Parametern bzw. Variablen, die zu Beginn vereinbart wurden. Das Element `<subreport>` ermöglicht die Einbindung von XML-Templates in XML-Templates. Daher werden `<subreports>` für die Ausgabe von Bausteinen und Freitext-Feldern (ProSAB-Bestandteil) angewendet. Aufgrund der Mehrfachverwendung von bestimmten ProSAB-Templates werden diese als Bausteine bezeichnet. Freitext-Felder stellen Texteditoren dar, die nach dem Parsen des ProSAB-Templates in der Bearbeitungsansicht von ProSAB ausgegeben werden. Sie können durch den Sachbearbeiter mit Text bzw. Tabellen gefüllt werden. Anschließend wird der eingegebene Text

---

als Subreport in den JasperPrint eingefügt, woraus anschließend das Schriftstück im PDF-Format hervorgeht.

Elemente, wie `<textField>` und `<subreport>`, die in Sektionen aufgerufen werden, besitzen das Kindelement `<reportElement>`. Dieses enthält die Pflichtattribute *x*, *y*, *width* und *height*. Sie dienen der Angabe der Koordinaten (in Pixel). Der größte kumulierte *y* und *height* Wert eines Elements im `<band>` ergibt die Mindesthöhe des Bands (siehe Abbildung 5).

Die Sektion `<detail>` wird nur in Bausteinen verwendet, insbesondere im Zusammenhang mit der grafischen Darstellung von Tabellen. Das Element `<detail>` verfügt analog den anderen Sektionen über das Kindelement `<band>`, welches die Elemente `<textField>` und weitere grafische Elemente enthält. Im Zusammenwirken mit `<tableFilter>` ergibt sich die Möglichkeit, Listenobjekte entsprechend der Objektanzahl zeilenweise in Tabellen auszugeben.

Der nachfolgende Aufruf der Sektion `<summary>` bezweckt eine abschließende Zeile, die z.B. der Ausgabe von Spaltensummen dient.

Die Sektionen `<pageHeader>` und `<pageFooter>` dienen der optionalen Ausgabe von Dokumentüberschriften und Seitenzahlen, jeweils am Seitenanfang bzw. -ende.

### **2.3 XML-Editoren als WYSIWYG-Tools**

WYSIWYG bedeutet wortwörtlich „Was Du siehst, ist, was Du bekommst“. Dieser Begriff entstammt dem „Xerox Palo Alto Research Center“<sup>23</sup> (Xerox PARC)<sup>24</sup>. Im Jahre 1974 entwickelte diese Institution den ersten WYSIWYG-Editor „Bravo“. Basierend auf diesem Programm entstand das allseits bekannte Textverarbeitungsprogramm Microsoft® Word.

Die Besonderheit von WYSIWYG-Editoren besteht darin, dass sie eine Layout-Ansicht anbieten, die bereits bei der Erstellung des Dokuments die endgültige Ausgabe wiedergibt. Anhand dieser hat der Autor bereits eine Vorstellung über das endgültige Layout des Dokuments und kann somit im voraus geeignete Formatierungen am Text vornehmen. Durch WYSIWYG-fähige Editoren wird die Erstellung von Dokumenten enorm vereinfacht.

---

<sup>23</sup> Xerox PARC ist eine Forschungseinrichtung des Kopiererherstellers Xerox.

<sup>24</sup> vgl. PARC (2009)

### 2.3.1 Allgemeiner Aufbau von XML-Editoren

Die Erstellung von XML-Dokumenten erfolgte in den vergangenen Jahren vor allem mit XML-Editoren, die lediglich über eine Textlayout-Ansicht verfügten (siehe Abbildung 6).

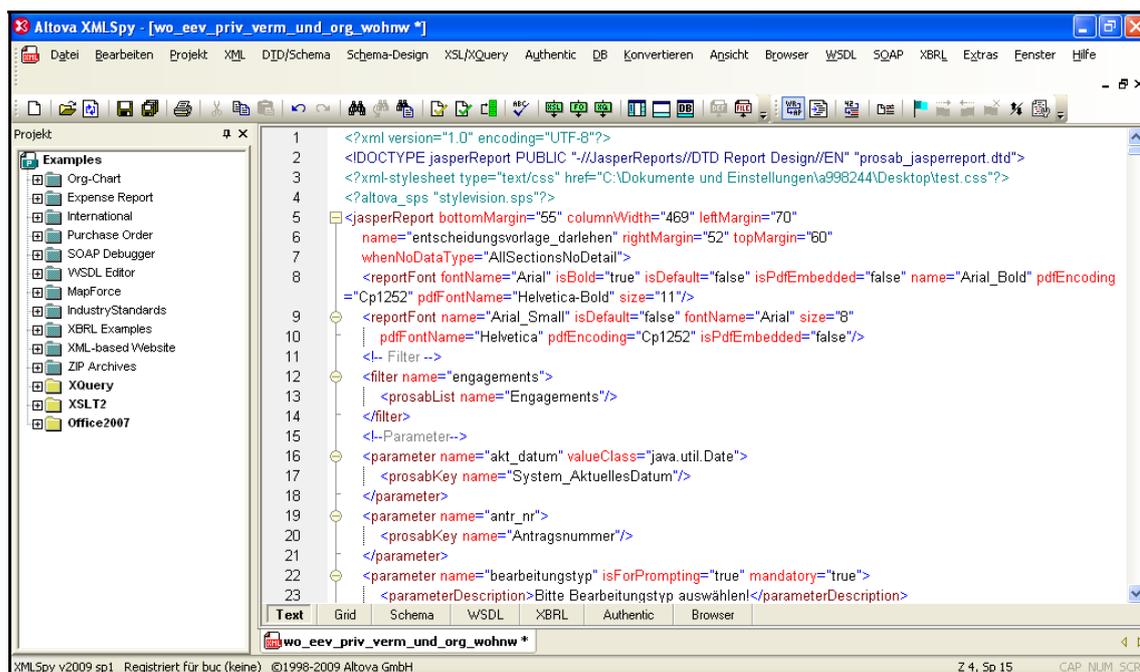


Abbildung 6: Textlayout-Ansicht im XML-Editor XMLSpy von Altova

Die Textlayout-Ansicht dient der Eingabe von XML-Code. Der Entwickler gibt manuell die Tags und deren Inhalt ein. Ausgereifte XML-Editoren verfügen über mehrere Funktionen, die den Entwickler bei der Erstellung von XML-Dokumenten unterstützen. Hierbei sind kontextsensitive Eingabevorschläge zu erwähnen, die entsprechend der DTD bzw. XSD Tags oder Attributwerte vorgeben. Per Mausklick bzw. Bestätigung mit der Enter-Taste der Tastatur kann das ausgewählte Element eingefügt werden. Jedoch verlangt das Arbeiten in der Textlayout-Ansicht viel Zeit, Konzentration und erweiterte Kenntnisse des Entwicklers im Umgang mit XML. Dieser Mehraufwand soll durch neue Bearbeitungsansichten beseitigt werden. Daher verfügen XML-Editoren über neue Ansichten wie Gridlayout-Ansicht und WYSIWYG-Layout-Ansicht.

In der Gridlayout-Ansicht wird das XML-Dokument in tabellarischer Form dargestellt. Alle Elemente sind strukturiert aufgelistet (siehe Abbildung 7).

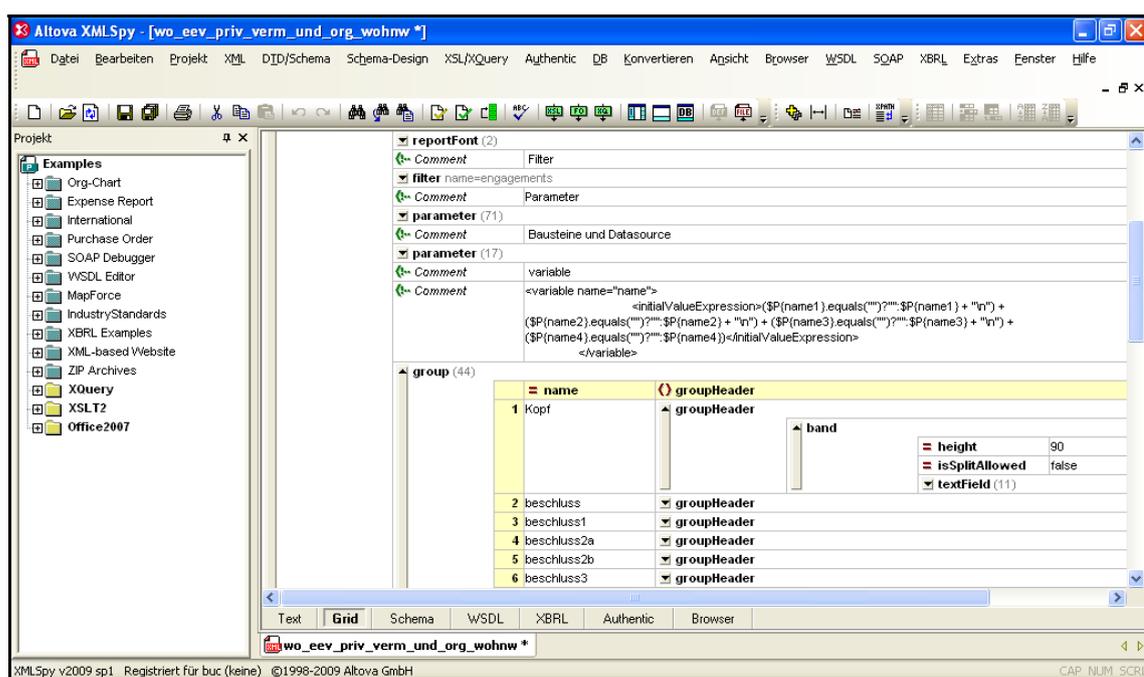


Abbildung 7: Gridlayout-Ansicht im Editor XMLSpy von Altova

Mit Hilfe einer Kontexterweiterung können die Attribute und Kindelemente detailliert dargestellt werden. Diese Layout-Ansicht ermöglicht ebenfalls die Bearbeitung des XML-Dokumentes, neue Elemente und Attribute können hinzugefügt bzw. entfernt werden. Die Eingabe der dazugehörigen Werte erfolgt dabei durch Mausklick in die jeweilige Zelle und manuelle Eingabe der Daten.

Die WYSIWYG-Layout-Ansicht stellt nur den Text sowie die Bilder des XML-Dokuments dar (siehe Abbildung 8).

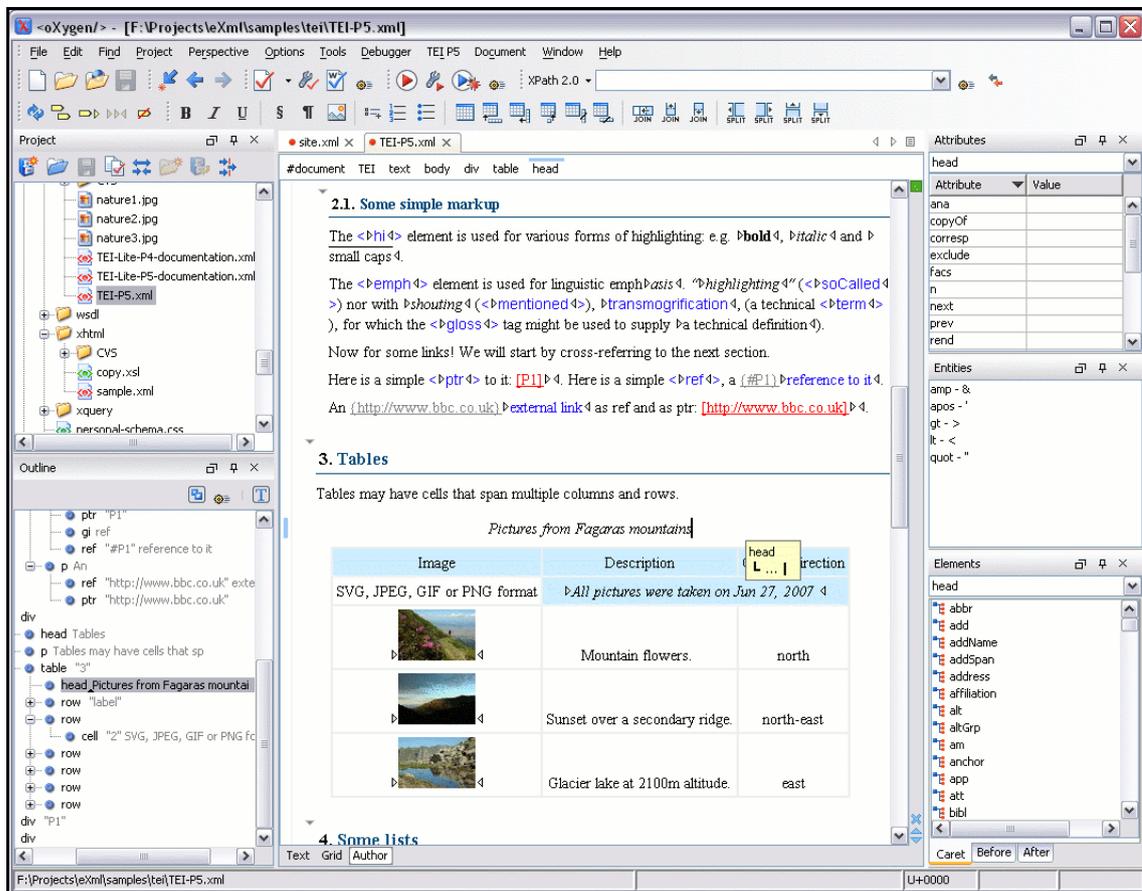


Abbildung 8: WYSIWYG-Layout-Ansicht in Oxygen von SyncRO Soft<sup>25</sup>

Die Bearbeitung des XML-Dokuments ist weiterhin möglich. Für die Eingabe weiterer Daten ist allerdings das Einblenden der Tags in Form von Symbolen empfohlen. Ebenfalls wie in der Textlayout-Ansicht sind Elemente und Attribute durch Tags zu vereinbaren. Anschließend kann der Inhalt zwischen den Tags eingefügt werden. Die Angabe der Werte zu den Attributen erfolgt meistens in einem zusätzlichen Bearbeitungsfenster. Es besteht auch die Möglichkeit, jederzeit zwischen den Bearbeitungsansichten zu wechseln und die Eingabe der Werte in der jeweiligen Layout-Ansicht fortzuführen. Die Verwendung der WYSIWYG-Layout-Ansicht ist allerdings stets an ein Stylesheet<sup>26</sup> gebunden. Dieses kann mittels Cascading Style Sheets oder Extensible Stylesheet Language (XSL) bereitgestellt werden.

<sup>25</sup> OXYGEN (2009)

<sup>26</sup> Formatvorlage für die Präsentation von Daten.

### 2.3.2 Verwendung der WYSIWYG-Layout-Ansicht bei XML-Editoren

Das Anzeigen und Bearbeiten von XML-Dokumenten in der WYSIWYG-Layout-Ansicht erfordert die Formatierung des Inhalts. Hierzu müssen Formatierungsinformationen auf das XML-Dokument angewendet werden<sup>27</sup>. CSS ist eine Stylesheet Sprache, die der Präsentation von Markup-Sprachen dient. Dabei ermöglicht CSS die genaue Beschreibung von Elementen durch Zuordnung von Eigenschaften. Durch die Definition der Elemente wird die Darstellung des XML-Dokuments vorgegeben. In den Ansichten wird dabei nur der Inhalt der Markups des XML-Dokuments formatiert ausgegeben. Die Formatierungsregeln werden in einer separaten Datei festgehalten. Der Aufruf der CSS-Datei geschieht im Prolog des XML-Dokuments. Dadurch ist es möglich, den Inhalt des XML-Dokuments in der WYSIWYG-Layout-Ansicht sowie im Web-Browser darzustellen. Die Anzeige im Web-Browser dient vor allem der Seitenvorschau des XML-Dokuments.

Die Formatierung von XML-Dokumentinhalten und die Erstellung neuer Dokumente in der WYSIWYG-Layout-Ansicht kann ebenso mit Hilfe von XSLT vorgenommen werden. Sie ist Bestandteil von XSL und eine Empfehlung des W3C. Die flexible und leistungsfähige Sprache XSLT ermöglicht die Formatierung und Umwandlung von XML-Dokumenten in ein anderes Format wie HTML oder PDF.

XSLT-Stylesheets enthalten Template-Regeln, die auf Elemente (Knoten) eines XML-Dokuments angewendet werden. Der Aufbau eines XML-Dokuments entspricht einem Baum, der aus mehreren Knoten besteht. Mittels der XPath-Syntax kann XSLT gezielt Elemente bzw. Knoten des XML-Eingabedokuments auswählen, die dann in das Ausgabedokument kopiert oder anderweitig verarbeitet werden<sup>28</sup>. Die Auswahl des Elements im Eingabedokument erfolgt mit Hilfe der Template-Regel im Stylesheet. Sie wird als *xsl:template-Element* bezeichnet. Dieses Element enthält das Attribut *match* mit einem Muster, das im Eingabedokument mit den Elementen verglichen wird. Wird eine Übereinstimmung zwischen Element und Muster gefunden, so wird die Anweisung der Template-Regel auf das Element des Eingabedokuments angewandt. Entsprechend der Anweisung werden anschließend die Daten in das Ausgabedokument übernommen. Die Daten können zum Beispiel Text sein, der in der Anweisung definiert ist oder Inhalt aus dem Element des Eingabedokuments.

---

<sup>27</sup> vgl. HAROLD, E. R./MEANS, W. S. (2005), S. 222

<sup>28</sup> vgl. Ebenda, S. 164

---

## **2.4 Lösungsansatz zur Untersuchung der XML-Editoren**

Anhand der Vorgaben konzentriert sich die Untersuchung von XML-Editoren ausschließlich auf Softwareprodukte, die WYSIWYG für XML unterstützen. Eine realistische Grundlage zur Beurteilung des Einsatzes von Editoren sowie die Auswahl potenzieller Softwareprodukte für die SAB bietet eine Evaluation. In diesem Kontext werden eine Marktanalyse der Produkte, Auswahl und Bewertung der Anwendungen vorgenommen. Mit ausgewählten Softwareprodukten werden anschließend praktische Tests vorgenommen. Dazu wird prototypisch das XML-Template EEV entwickelt, das gebrauchstypische Elemente wie ProSAB-Keys und -Listen sowie Bausteine enthält. Die anschließende Auswertung der gewonnenen Erkenntnisse ermöglicht eine sachgerechte Entscheidung über die Verwendung von WYSIWYG-fähigen XML-Editoren und dem möglichen Softwareprodukt.

### 3 Evaluation von WYSIWYG-fähigen XML-Editoren

Die Softwareevaluation ist ein Vorgang, der Analyse, Bewertung und Beurteilung von Software umfasst<sup>29</sup>. Basierend auf einer festgelegten Referenzskala und Kriterien erfolgt die Beurteilung der Software. Voraussetzung für die Beurteilung sind Analyse und Bewertung der Produkte anhand der Kriterien.

#### 3.1 Konzeption

In Anbetracht der fortschreitenden Entwicklung von XML-Editoren in Hinsicht auf WYSIWYG ist die SAB interessiert, das Potenzial dieser Editoren zu ermitteln sowie einen möglichen Einsatz im Haus zu erwägen.

Der Einsatz eines neuen Editors soll wirtschaftliche Vorteile erbringen. Die Erstellung von XML-Templates für ProSAB mittels einer WYSIWYG-Layout-Ansicht führt eventuell zu Zeitersparnissen. Demzufolge werden weniger Personentage für die Erstellung von XML-Templates benötigt. Zusätzlicher Effekt ist die kürzere Einarbeitungszeit für Mitarbeiter, die im Umgang mit XML-Editoren sowie der Erstellung von XML-Templates nicht vertraut sind.

##### 3.1.1 Ist-Analyse

Die Erstellung von XML-Templates erfolgt bisher mit dem XML-Editor „XMLSpy Professional Edition“ von Altova. Dieser leistungsstarke Editor bietet vier Bearbeitungsansichten. Die Textlayout-Ansicht stellt eine Ansicht davon dar, sie ermöglicht die Eingabe des XML-Codes durch den Entwickler. Dabei werden Markups, deren Inhalt und Attribute in unterschiedlichen Farben hervorgehoben. Während der Eingabe „On the fly“ wird der XML-Code auf Wohlgeformtheit geprüft und gegen eine vorgegebene DTD validiert. Eingaben des Entwicklers werden durch XMLSpy unterstützt, indem eine Kontextauswahl an verschiedenen Stellen im Code zur Verfügung steht, wie z.B. Auswahlliste mit Elementen, Attributen und deren Werten, sofern die Informationen in der DTD vorhanden sind.

Eine weitere Bearbeitungsansicht ist das Gridlayout, die Ansicht stellt das XML-Dokument tabellarisch dar. Hierbei werden Elemente untereinander aufgelistet. Attribu-

---

<sup>29</sup> vgl. STOWASSER, S. (2006), S. 28

---

te und Kindelemente werden strukturiert in Zellen ausgegeben. Die Gridlayout-Ansicht ermöglicht die Bearbeitung des XML-Dokuments, indem analog zur Textlayout-Ansicht diverse Kontexthilfen zur Verfügung stehen. Vorgenommene Veränderungen haben umgehende Wirkung auf den erstellten XML-Code und können in der Textlayout-Ansicht betrachtet werden.

XMLSpy unterstützt neben DTD ebenfalls XSD. Gegenwärtig wird für ProSAB-Templates eine eigene DTD verwendet, die auf der vorgegebenen DTD von JasperReports basiert.

XMLSpy unterstützt außerdem zahlreiche Funktionen, wie Suchen und Ersetzen, „Drag and Drop“ und die Ausrichtung des XML-Codes in der Textlayout-Ansicht.

### **3.1.2 Soll-Konzeption**

Die Ist-Analyse ergab bereits zahlreiche Funktionen, die XMLSpy erfüllt. Besonders in der Textlayout-Ansicht stehen Funktionen zur Verfügung, die eine benutzerfreundliche Erstellung von Templates ermöglichen. Daher besteht die Anforderung an zu untersuchende WYSIWYG-fähige XML-Editoren, dass sie über eine Textlayout-Ansicht verfügen, die den bereits bekannten Funktionen nachkommt. Sobald eine Eingabe innerhalb der WYSIWYG-Layout-Ansicht nicht zu tätigen ist, muss alternativ eine Textlayout-Ansicht zur Verfügung stehen. Folgende Funktionen muss sie erfüllen: Überprüfung des XML-Codes auf Wohlgeformtheit und Validierung dessen gegen die vorgegebene DTD, farbliches Hervorheben von Elementen und Attributen im Template, Markierung von Fehlern und kontextsensitive Empfehlung von Eingaben durch Auswahllisten. Weitere Funktionen, wie „Drag and Drop“, Suchen und Ersetzen werden ebenfalls erwartet.

Außerdem sollen die Editoren über eine WYSIWYG-Layout-Ansicht verfügen, die äquivalent zur Textlayout-Ansicht das Erstellen von ProSAB-Templates ermöglicht. Eine bevorzugte Technik wäre dabei, das Einfügen von Elementen per „Drag and Drop“ aus einer Liste von Elementen, die der zugewiesenen DTD entsprechen. Gleichzeitig sind Eingaben, die in der WYSIWYG-Layout-Ansicht getätigt werden, automatisch in der Textlayout-Ansicht zu realisieren. Diesbezüglich wäre es vorteilhaft, wenn beim Einfügen von Kindelementen in der WYSIWYG-Layout-Ansicht, entsprechend der DTD, die Hierarchie der Elemente selbstständig in der Textlayout-Ansicht abgebildet wird. Im Anschluss sind Text bzw. Keys in der WYSIWYG-Layout-Ansicht einzuge-

ben. Ein weiterer Nutzen entsteht für den Entwickler, falls die Möglichkeit besteht, den eingegebenen Text zu markieren und ihn mittels Funktionen aus der Symbolleiste zu formatieren. Auf dieselbe Art kann dem eingegebenen Text ein Element zugewiesen werden. Die Formatierung ist ebenfalls im XML-Code durch den Editor darzustellen. Entsprechend der Textlayout-Ansicht sind die vorgenommenen Eingaben in der WYSIWYG-Layout-Ansicht gegen die zugewiesene DTD zu validieren. Dabei sind erkannte Fehler zu kennzeichnen. Für die Eingabe von Attributen zuzüglich der Werte, ist ein separates Eingabefenster vorstellbar.

Zusätzliche Bearbeitungsansichten (z.B. Gridlayout-Ansicht) sind erwünscht, jedoch werden sie nicht zwingend erfordert. Weiterhin ist die Bereitstellung einer Seitenansicht erwünscht bzw. eine Zoommöglichkeit in der WYSIWYG-Layout-Ansicht, wodurch eine Gesamtansicht auf das zu erstellende Schriftstück gegeben ist.

### 3.1.3 Aufstellung von Bewertungskriterien

Basierend auf den ermittelten Informationen der Ist-Analyse erfolgte die Formulierung von Anforderungen an die zu untersuchenden WYSIWYG-fähigen XML-Editoren. In diesem Zusammenhang bestand eine Wechselwirkung zwischen den erstellten Anforderungen und der anschließenden Evaluation der Editoren<sup>30</sup>. Aufgestellte Kriterien an die zu untersuchenden XML-Editoren bildeten die Basis für die Evaluation. Während der Evaluation der Softwareprodukte besteht allerdings die Möglichkeit, dass neue Erkenntnisse gewonnen werden, die eine Änderung der Bewertungskriterien erfordern.

Kriterien, die der Auswahl von Softwareprodukten dienen, werden Bewertungskriterien genannt. Darunter werden ebenfalls Bewertungskriterien aufgelistet, die zwingend erfüllt sein müssen. Diese Art von Kriterien wird als „K.O.-Kriterium“ bezeichnet. Sobald ein K.O.-Kriterium durch einen XML-Editor nicht erfüllt ist, muss das Softwareprodukt aus der Bewertung ausgeschlossen werden<sup>31</sup>. Diese Verfahrensweise ermöglicht eine frühzeitige Eingrenzung der Menge an zu evaluierende WYSIWYG-fähige Editoren.

Für die Durchführung der Evaluation wurde ein eigener Kriterienkatalog erstellt, der den spezifischen ProSAB- und SAB-Anforderungen gerecht wurde. Daher wurden durch den Autor, in Zusammenarbeit mit der Gruppe Anwendungsadministration der Abteilung Informationstechnologie, Bewertungskriterien aufgestellt. Neben den indivi-

---

<sup>30</sup> vgl. BECKER, J./VERING, O./WINKELMANN, A. (2007), S. 75

<sup>31</sup> vgl. Ebenda, S. 121

duellen Anforderungen an die Produkte waren Aspekte, wie Softwarequalität und Softwareergonomie von Bedeutung. Aus diesem Grund wurden bezüglich der Qualität und Ergonomie, Normen der International Organization for Standardization (ISO) einbezogen. Sie stellen Regeln dar, die von Expertengremien erstellt wurden und international anerkannt sind. Aufgrund dessen, dass es sich um Standards (z.B. für die Mensch-Computer Interaktion) handelt, sind sie allgemein formuliert und nicht auf spezielle Anwendungen ausgerichtet.

Die aufgestellten Bewertungskriterien wurden zur besseren Übersicht thematisch in Gruppen zusammengefasst, sie werden nachfolgend aufgelistet. Der vollständige Erhebungsbogen mit Hersteller- und Produktinformationen zu den aufgestellten Kriterien ist im Anhang (Anlage 1) der Arbeit einzusehen. Die erste Gruppe enthält allgemeine Informationen zu den Anbietern und Produkten, sie fließt daher nicht in die Bewertung ein.

In Gruppe 2 sind messbare Kriterien zu den Anbietern und Produkten enthalten. Hierbei ist der Benutzersupport von hoher Bedeutung. Bei auftretenden Problemen wird eine schnelle, umfangreiche Unterstützung erwartet. Weiterhin sind der Preis pro Lizenz und die Referenzen des Anbieters von Bedeutung.

<b>Angaben zum Produkt und Anbieter</b>			
<b>Nr.</b>	<b>Kriterium</b>	<b>K.O. Kr.</b>	<b>Beschreibung</b>
2.1	Bekanntheitsgrad des Anbieters und Produktes		Der Bekanntheitsgrad lässt darauf schließen, dass der Anbieter am Markt etabliert ist und das Softwareprodukt ständige Weiterentwicklung erfährt.
2.2	Leistungsfähigkeit des Unternehmens		Die Größe des Anbieters vermittelt einen ersten Eindruck über die Kapazität des Unternehmens. Dazu werden die Mitarbeiteranzahl und der Jahresumsatz betrachtet.
2.3	Anzahl der Lizenzen		Die Anzahl der verkauften Lizenzen verschafft einen Überblick über den Marktanteil des Anbieters.
2.4	Release-Häufigkeit		Die Release-Häufigkeit gibt Aufschluss darüber, in welchen Zeiträumen neue Versionen bereitgestellt werden. Weiterhin könnte sie ein Indiz für die Reife der Software sein.
2.5	Referenzen		Referenzen im Bank- und Versicherungsgewerbe sind Indiz dafür, dass die Software mit hoher Wahrscheinlichkeit den Anforderungen der SAB entspricht.
2.6	Benutzersupport, Wartung		Der Nutzer erhält bei Problemen Unterstützung durch den Softwareanbieter. Wartungsverträge ermöglichen darüber hinaus größtenteils Updates und neue Versionen.

2.7	Add-ons für das Produkt		Add-ons sind überwiegend kostenlose Softwareerweiterungen, die zusätzliche Programmfunktionen wie Rechtschreibprüfung und Konfiguration der Software ermöglichen.
2.8	Preis pro Produkt		Preis- bzw. Lizenzkosten, die durch den Erwerb eines Softwareproduktes entstehen.

**Tabelle 1: Kriterien zum Produkt und Anbieter**

Die Gruppe 3 enthält Kriterien zu Systemeigenschaften, welche durch die Editoren zu erfüllen sind. Dabei ist ein K.O.-Kriterium die Unterstützung des Betriebssystems Windows.

<b>Allgemeine Systemeigenschaften</b>			
<b>Nr.</b>	<b>Kriterium</b>	<b>K.O. Kr.</b>	<b>Beschreibung</b>
3.1	Unterstützung der Betriebssysteme Windows 2000, XP, Vista, evtl. 7	ja	Die voraussichtliche Umstellung der Betriebssysteme in der SAB, von Windows 2000 auf ein aktuelles Windows Betriebssystem, erfordert die Aufwärtskompatibilität des
3.2	Applikation	ja	Das Softwareprodukt ist als eigenständige Applikation auf Einzelplatzrechnern auszuführen, die Verwendung von Thin-Clients in Verbindung mit einem Server ist nicht erwünscht.
3.3	Erstellung von XML-Dokumenten nach W3C Spezifikation		Das W3C erstellt Standards, die in XML-Dokumenten zu befolgen sind. Die Vorgaben sind durch das Softwareprodukt einzuhalten.
3.4	Unterstützung von JasperReports		Die Arbeit mit JasperReports spezifischen Verarbeitungsbestandteilen, wie Sektionen ( <i>PageHeader</i> , <i>Group</i> ) und Subreports ist durch die Software zu ermöglichen.

**Tabelle 2: Kriterien zu den allgemeinen Systemeigenschaften**

In Tabelle 3 sind Kriterien für die Benutzeroberfläche der XML-Editoren aufgelistet, sie bilden die Gruppe 4. Die Bearbeitungsansichten Textlayout und WYSIWYG-Layout repräsentieren hierbei zwei K.O. Kriterien, die unerlässlich für die Untersuchung der Editoren waren.

<b>Benutzeroberfläche</b>			
<b>Nr.</b>	<b>Kriterium</b>	<b>K.O. Kr.</b>	<b>Beschreibung</b>
4.1	Textlayout-Ansicht	ja	Es besteht die Möglichkeit, den XML-Code in der Textlayout-Ansicht zu bearbeiten.
4.2	Gridlayout-Ansicht (Baumansicht)		Die Bearbeitung des XML-Dokuments ist in einer tabellarischen Ansicht möglich.
4.3	WYSIWYG-Layout-Ansicht	ja	XML-Dokumente werden in einer speziellen Ansicht erstellt und bearbeitet.
4.4	Menüsprache - deutsch - englisch		Die Beschriftung des Menüs sowie der Dialog zwischen Nutzer und Softwareprodukt erfolgt vornehmlich in deutscher, alternativ in englischer Sprache.
4.5	Seitenvorschau		Das Softwareprodukt ermöglicht eine Vorschau auf das in Bearbeitung befindende XML-Dokument.

**Tabelle 3: Kriterien zur Benutzeroberfläche**

Nachfolgend werden in Tabelle 4 Kriterien zur Dateneingabe und -bearbeitung aufgeführt. Sie enthalten Anforderungen, wie Unterstützung von DTD, die zwingend zu erfüllen sind. Außerdem werden Funktionen (z.B. kontextsensitive Eingabemöglichkeiten) erwünscht, die für eine komfortable Bearbeitung von Templates sorgen.

<b>Entwurf, Dateneingabe und -bearbeitung</b>			
<b>Nr.</b>	<b>Kriterium</b>	<b>K.O. Kr.</b>	<b>Beschreibung</b>
5.1	Unterstützt DTD und XSD	ja	Das Softwareprodukt unterstützt die Verwendung von DTD und XML-Schema Definition.
5.2	kontextsensitive Eingabemöglichkeiten		In Form von Auswahllisten oder zusätzlichen Fenstern werden kontextsensitiv die zum Einfügen erlaubten Elemente und Attribute angezeigt.
5.3	kontextsensitive Bearbeitungshilfen		Während der Bearbeitung des XML-Dokuments werden kontextsensitiv Eingaben vorgeschlagen bzw. Tags ergänzt. Zu eingefügten Start-Tags wird automatisch ein End-Tag hinzugefügt.
5.4	Code-Snippets		Es besteht die Möglichkeit, vorhandene Code-Stücke über das Menü einzufügen bzw. diese Code-Stücke selbst zu definieren.
5.5	Standardfunktionen		Die Softwareprodukte ermöglichen die Verwendung von Standardfunktionen, wie Short Cuts, Kopieren und Einfügen, Ausschneiden, Suchen und Ersetzen, „Drag and Drop“.
5.6	Unicode und andere Zeichensätze	ja	Die Verwendung des Unicode-Zeichensatzes wird unterstützt, optional werden weitere Zeichensätze angeboten.

5.7	Einfügen von Grafiken und Tabellen		Es ist möglich Grafiken und Tabellen in das XML-Dokument einzufügen.
5.8	Versionierung von Templates		Das Softwareprodukt unterstützt die Versionsverwaltung von XML-Dokumenten. Die Verwaltung der Dokumente erfolgt mittels Subversion.
5.9	Struktur und Umfang des generierten XML-Codes		Der automatisch erzeugte XML-Code wird hinsichtlich des Umfangs bzw. Struktur der Elemente überprüft.

**Tabelle 4: Kriterien zur Dateneingabe und -bearbeitung in den Editoren**

Die Gruppe 6 enthält Kriterien zu Prüfroutinen, die durch Editoren in den jeweiligen Layout-Ansichten entsprechend bereitzustellen sind.

<b>Prüfroutinen</b>			
<b>Nr.</b>	<b>Kriterium</b>	<b>K.O. .Kr.</b>	<b>Beschreibung</b>
6.1	Validierung des XML-Templates	ja	Das Softwareprodukt validiert den XML-Code des geöffneten XML-Dokuments gegen die vorgegebene DTD.
6.2	„On the fly“-Validierung		Während der Eingabe von XML-Code validiert das Softwareprodukt selbstständig den XML-Code gegen die vorgegebene DTD.
6.3	Validierung ein- und ausschalten		Es besteht die Möglichkeit, dass Softwareprodukte das Speichern nicht valider XML-Dokumente verhindern. Unter diesen Umständen ist die Deaktivierung der Funktion hilfreich, damit das Dokument trotzdem gespeichert wird.
6.4	Prüfung auf Wohlgeformtheit	ja	Das Softwareprodukt überprüft das in Bearbeitung befindliche XML-Dokument auf Wohlgeformtheit.
6.5	Hervorhebung von Syntaxfehlern in der Textlayout-Ansicht		Sofern im XML-Dokument die Syntax fehlerbehaftet ist, erfolgt eine farbliche Kennzeichnung des betroffenen XML-Codestücks.
6.6	Rechtschreibprüfung		Der eingefügte Text kann in den Layout-Ansichten durch das Softwareprodukt einer Rechtschreibprüfung unterzogen werden.

**Tabelle 5: Kriterien zu den Prüfroutinen**

In der Gruppe Softwarequalität erfolgt die Bewertung der Produktfunktionen und -leistung. Die Norm ISO/IEC 9126 gilt speziell der Bewertung von Softwareproduktqualität. Nachfolgend sind die Qualitätsmerkmale der Norm aufgeführt.

<b>Softwarequalität</b>			
<b>Nr.</b>	<b>Kriterium</b>	<b>K.O. Kr.</b>	<b>Beschreibung</b>
7.1	Funktionalität		Bereitstellung der geforderten Funktionen durch das Softwareprodukt.
7.2	Zuverlässigkeit		Die Software kann ein bestimmtes Leistungsniveau über einen festgelegten Zeitraum aufrechterhalten.
7.3	Benutzbarkeit		Aufwand, der zur Nutzung der Software durch den Benutzer zu erbringen ist.
7.4	Effizienz		Verhältnis zwischen dem Leistungsniveau der Software und den eingesetzten Betriebsmitteln.
7.5	Änderbarkeit		Aufwand, der zur Durchführung von Veränderungen an der Software erforderlich ist.
7.6	Übertragbarkeit		Eignung der Software aus einer Umgebung in eine andere übertragen zu werden, in Bezug auf Hardware und Software.

**Tabelle 6: Kriterien zur Softwarequalität**

Die letzte Gruppe enthält Bewertungskriterien zur Ergonomie der Softwareprodukte, dazu wird die Norm EN ISO 9241 angewandt. Sie beschreibt Richtlinien der Interaktion zwischen Mensch und Computer. Im Unterpunkt 110 „Grundsätze der Dialoggestaltung“ werden die Grundsätze der Bewertung von Schnittstellen beschrieben.

<b>Softwareergonomie</b>			
<b>Nr.</b>	<b>Kriterium</b>	<b>K.O. Kr.</b>	<b>Beschreibung</b>
8.1	Aufgabenangemessenheit		Dialoggestaltung, angemessene Darstellung der Informationen in Bezug auf die Aufgabenerfüllung.
8.2	Selbstbeschreibungsfähigkeit		Bereitstellung von Informationen, damit für den Nutzer ersichtlich ist, in welchem Dialog und an welcher Stelle im Dialog er sich befindet.
8.3	Steuerbarkeit		Der Nutzer kann den Dialog zwischen ihm und dem Softwareprodukt steuern, indem er ihn startet, seine Richtung und Geschwindigkeit beeinflusst.
8.4	Erwartungskonformität		Der Dialog muss dem Nutzungskontext und anerkannten Konventionen entsprechen.
8.5	Fehlertoleranz/ Fehlerrobustheit		Das Softwareprodukt verhindert fehlerhafte Eingaben, sofern diese erfolgt sind, ist der Aufwand zur Beseitigung der Fehler minimal.
8.6	Individualisierbarkeit		Die Darstellung von Informationen kann an individuelle Bedürfnisse angepasst werden.
8.7	Lernförderlichkeit		Der Nutzer erhält Hilfestellungen zu Regeln und Konzepten, die das Softwareprodukt verwendet, dadurch erlangt er Hintergrundwissen.

**Tabelle 7: Kriterien zur Softwareergonomie**

### 3.1.4 Erhebung der Produktinformationen

Auf der Grundlage des Soll-Konzepts erfolgte eine Marktsichtung. Sie verschaffte einen Überblick über potenzielle Softwareanbieter. Die Ermittlung möglicher Softwareanbieter erfordert Kenntnis über verfügbare Standardlösungen sowie eine erste einfache und klare Trennung zwischen potenziell geeigneten und eher ungeeigneten Systemen.<sup>32</sup> Eine Übersicht an Softwareanbietern kann anhand von Branchen- und Fachpublikationen, Softwarekataloge und Unternehmensberatern gewonnen werden. Im Rahmen dieser Arbeit wurden zur Marktanalyse Online-Softwarekataloge<sup>33</sup> verwendet.

Softwarekataloge enthalten Kategorien, wodurch Softwareprodukte in verschiedene Anwendungsbereiche eingeordnet werden. Die Auswahl einer Kategorie verschafft eine rasche Marktübersicht über vorhandene Softwareprodukte. Zu jedem Produkt und Anbieter sind kurze Informationen aufgeführt sowie eine Verlinkung zur Webpräsentation des Anbieters.

Eine weitere Alternative sind Suchmaschinen, die ebenfalls online Softwarekataloge zur Verfügung stellen bzw. konventionelle Suchmaschinen (z.B. Google), die zahlreiche Produkte aufführen.

Webpräsentationen der Software-Anbieter stellen häufig eine Vielzahl von Informationen zu den angebotenen Softwareprodukten zur Verfügung. Darunter sind Benutzerhandbücher enthalten, die wichtige Hinweise zu den Produktfunktionen liefern.

Für die Erfassung der Anbieter- und Produktinformationen wurde ein Erhebungsbogen (siehe Anhang) erstellt, der auf den bereits erwähnten Bewertungskriterien für WYSIWYG-fähige XML-Editoren beruht. Die ermittelten Informationen der Anbieter-Webpräsentationen und Benutzerhandbücher wurden im Erhebungsbogen erfasst. Sie bildeten eine wichtige Grundlage für die spätere Auswertung der Produktinformationen. Im Rahmen der Erhebung von Produktinformationen wurden allerdings nur jene Bewertungskriterien betrachtet, die messbar bzw. mit den Wahrheitswerten Ja/Nein zu beantworten waren. Hierfür wurde die Verfügbarkeit der benötigten Informationen vorausgesetzt. Informationen zu Bewertungskriterien bezüglich der Benutzbarkeit der WYSIWYG-fähigen Editoren wurden im Zusammenhang mit Produkttests erfasst.

---

<sup>32</sup> vgl. BECKER, J./VERING, O./WINKELMANN, A. (2007), S. 82

<sup>33</sup> SOFTWARE, SOFTGUIDE, COMPUTERWOCHE

## 3.2 Vorauswahl

Nachdem neun XML-Editoren einschließlich der Produktinformationen in Form eines Erhebungsbogens bereitstanden, war ersichtlich, dass verschiedene Bewertungskriterien durch alle Editoren gleichermaßen erfüllt bzw. nicht erfüllt wurden oder keine Angaben vorhanden waren. Diese Bewertungskriterien leisteten keinen Beitrag zur Hervorhebung von einzelnen Softwareprodukten. Aus diesem Grund wurden sie aus dem Erhebungsbogen nicht in den Bewertungsbogen (siehe Anhang, Anlage 2) übernommen.

Durch den Bewertungsbogen erfolgte eine Verringerung der Anzahl von WYSIWYG-fähigen Editoren, die einem praktischen Test unterzogen wurden. Diese Maßnahme war hinsichtlich des Zeitaufwandes der Tests notwendig. Zunächst wurden in einem Bewertungsbogen nur Bewertungskriterien der Gruppen „Produkt- und Anbieterinformationen“ sowie „Allgemeine Systemeigenschaften“ analysiert.

Damit die einzelnen XML-Editoren vergleichbar waren und eine engere Auswahl getroffen werden konnte, erfolgte eine Gewichtung der einzelnen Bewertungskriterien. Weiterhin war jeder Editor hinsichtlich der Erfüllung einzelner Bewertungskriterien zu bewerten.

Die Festlegung einer Gewichtung oder Präferenzordnung für Bewertungskriterien erfolgt in der Regel mittels Methoden der Entscheidungstheorie und Nutzwertanalyse<sup>34</sup>.

In der vorliegenden Arbeit erfolgte die Auswahl potenzieller Softwareprodukte durch eine Nutzwertanalyse. Die Nutzwertanalyse ermöglicht das Einbeziehen nicht quantifizierbarer, subjektiver Bewertungsgrößen in die Zielbildung<sup>35</sup>. Dazu wurde eine Skalierung der Bewertungskriterien vorgenommen. Jedes Bewertungskriterium einer Gruppe erhielt ein Gewicht in Form eines Prozentwertes, womit der Einfluss des Bewertungskriteriums innerhalb der Gruppe ausgedrückt wurde. Die Summe der Gewichte innerhalb jeder Gruppe ergibt einhundert Prozent. Weiterhin erfolgte eine Gewichtung der Gruppen untereinander, demzufolge hat jede Gruppe ebenfalls einen vorgegebenen Einfluss auf den prozentualen Gesamtwert eines jeden Softwareproduktes. Die Summe der Gruppengewichtung ergibt ebenso einhundert Prozent, dies ist der maximale Wert, der durch einen WYSIWYG-fähigen XML-Editor zu erreichen ist. Es folgte die prozentuale Bewertung jedes Editors, hinsichtlich der Erfüllung jedes Bewertungskriteriums, in der zweiten und dritten Gruppe. Die einzelnen Kriteriengewichte einer Gruppe wurden pro

---

<sup>34</sup> vgl. STOWASSER, S. (2006), S. 62

<sup>35</sup> vgl. PEPELS, W. (2006), S. 451

---

XML-Editor kumuliert und mit dem Gruppengewicht multipliziert. Das Ergebnis stellt einen Anteil vom prozentualen Gesamtwert eines Editors dar.

### **3.2.1 Ausgewählte XML-Editoren**

Es wurde bereits darauf hingewiesen, dass im Rahmen der ersten Produktauswertung, nur die Gruppen „Produkt- und Anbieterinformationen“ sowie „Allgemeine Systemeigenschaften“ analysiert wurden. Zu diesem Zeitpunkt war eine Beurteilung der XML-Editoren nur hinsichtlich der Bewertungskriterien dieser Gruppen möglich, da die benötigten Informationen durch Webpräsentationen des Anbieters sowie Benutzerhandbücher zur Verfügung standen. Nachfolgende Gruppen und Bewertungskriterien wurden anhand praktischer Tests eingeschätzt.

Angehts der praktischen Tests, die im Anschluss folgten und von hohem Zeitaufwand geprägt waren, erfolgte die Festlegung einer Auswahlgrenze. Diese begrenzte die Anzahl der zu untersuchenden Softwareprodukte. Vom Autor der Arbeit wurde festgelegt, dass die vier rangbesten Softwareprodukte näher untersucht wurden.

Zunächst wurde die Gruppe mit Angaben zum Produkt und Anbieter bewertet. Die Bewertung der Kriterien beruhte zum Teil auf subjektiven Wahrnehmungen. Der Bekanntheitsgrad des Anbieters und dessen Referenzen vertreten Bewertungskriterien, die subjektiv bewertet wurden. Ausschlaggebend für den Bekanntheitsgrad war die Präsenz des Anbieters und des Softwareproduktes am Markt, Erwähnung auf Websites, Foren, Katalogen und Suchmaschinen. Diese Informationen beruhten auf den Recherchen des Autors.

#### **Angaben zum Produkt und Anbieter**

Aus den Informationserhebungen des Autors ging das Unternehmen Altova mit dem Produkt XMLSpy als führender Anbieter hervor. Dies wird in Tabelle 8. ersichtlich, die einen Auszug des Bewertungsbogens abbildet. Es folgen die Softwareprodukte XMLmind, Oxygen und Serna.

Das Bewertungskriterium Referenz wurde anhand der aufgeführten Kunden, die jeder Anbieter aufwies, beurteilt. Dabei erhielten Anbieter, die auf einen hohen Anteil an Banken und Versicherungen verwiesen, einen höheren bzw. den maximalen Prozentwert.

Nr.	Angaben zum Produkt und Anbieter	K.O. Kr.	GW Kr.	GW Gr.	XMLmind	Serna	XMLSpy 2009	Oxygen XML Editor
2.1	Bekanntheitsgrad		15%	15%	10%	5%	15%	10%
2.5	Referenzen		20%		10%	10%	20%	15%
2.6	Benutzersupport		30%		25%	30%	25%	30%
2.7	Add-ons/Plugins		10%		10%	5%	10%	0%
2.8	Preis/Lizenz		25%		15%	10%	10%	10%
<b>Summe</b>			<b>100%</b>		<b>70%</b>	<b>65%</b>	<b>80%</b>	<b>65%</b>
<b>Gesamtgewicht in Gruppe</b>					<b>10,50%</b>	<b>9,75%</b>	<b>12%</b>	<b>9,75%</b>
Nr.	Allgemeine Systemeigenschaften	K.O. Kr.	GW Kr.	GW Gr.	XMLmind	Serna	XMLSpy 2009	Oxygen XML Editor
3.1	Unterstützt Windows Betriebssysteme	X	70%	10%	70%	70%	70%	70%
3.4	Unterstützt JasperReports		30%		0%	0%	0%	0%
<b>Summe</b>			<b>100%</b>		<b>70%</b>	<b>70%</b>	<b>70%</b>	<b>70%</b>
<b>Gesamtgewicht in Gruppe</b>				<b>7%</b>	<b>7%</b>	<b>7%</b>	<b>7%</b>	
<b>Summe 1. und 2. Gruppe</b>					<b>17,50%</b>	<b>16,75%</b>	<b>19%</b>	<b>16,75%</b>
<b>Rangfolge</b>					<b>2.</b>	<b>3.</b>	<b>1.</b>	<b>3.</b>

(GW-Gewicht; Kr. - Kriterium; Gr. - Gruppe)

**Tabelle 8: Bewertung der Anbieter und Systemeigenschaften für die ausgewählten Editoren**

Dem Benutzersupport gilt annähernd von allen Anbietern eine hohe Aufmerksamkeit. Diese Feststellung ist dem im Anhang beigefügten Erhebungsbogen zu entnehmen. Auf Anfrage offerieren Anbieter Unterstützung per E-Mail und Telefon. Weiterhin werden Wartungspakete angeboten, die Updates auf neue Versionen sowie Support (Dienstleistung) gewährleisten. Häufig beträgt der Dienstleistungszeitraum 1-2 Jahre. Die Bewertung des Kriteriums Benutzersupport entspricht dem angebotenen Leistungsumfang der Anbieter.

Ein Teil der Anbieter stellt außerdem kostenlose Add-ons (Erweiterung des Softwareprodukts) zur Verfügung. Die Installation der Add-ons ermöglicht zum Beispiel die Durchführung einer Rechtschreibprüfung oder die Bereitstellung von XSLT und XSLFO Prozessoren. Die Wertung des Kriteriums Benutzersupport basiert ebenfalls auf den gebotenen Leistungen der Anbieter.

Das letzte Kriterium der Gruppe gilt der Bewertung des Preises für ein Softwareprodukt. Hierbei wurde darauf geachtet, dass gleichwertige Softwareprodukte im Funktionsangebot einem Vergleich unterzogen wurden. Ein Großteil der Softwareprodukte entspricht im Funktionsumfang der Enterprise Version. Der Tabelle 8. ist zu entnehmen,

---

dass die Preise annähernd gleichwertig waren. Produkte die einen geringen Preis aufwiesen bzw. kostenlos waren erhielten eine höhere Bewertung.

### **Allgemeine Systemeigenschaften**

In der Gruppe „Allgemeine Systemeigenschaften“ ist das Bewertungskriterium „Unterstützung der Windows Betriebssysteme“ enthalten, dieses Kriterium wurde von allen Produkten erfüllt. Sofern ein Softwareprodukt diesem Kriterium nicht oder nur teilweise entsprechen würde, wäre der Bewertungswert gleich null.

Zuletzt ist das Kriterium „Unterstützung JasperReports“ enthalten, dieser Anforderung wurde kein Softwareprodukt gerecht. Daher wurden die XML-Editoren hinsichtlich dieses Bewertungskriteriums mit 0 % gewichtet.

Die Kumulierung der ermittelten Gewichte je Produkt über die genannten Gruppen ergab folgende Vorauswahl. Das Softwareprodukt XMLSpy 2009 von Altova erlangte im Rahmen der Bewertung den vorläufig höchsten Rang. Es folgt XMLmind von Pixware, an dritter Position stehen die Softwareprodukte Serna und Oxygen. Die aufgeführten Softwareprodukte wurden weiteren Überprüfungen und Tests unterzogen.

### 3.2.2 Aussortierte XML-Editoren

Im letzten Abschnitt erfolgte die Festlegung einer Auswahlgrenze für WYSIWYG-fähige XML-Editoren. Die durchgeführte Selektion führte zum Ausschluss mehrerer Softwareprodukte, die in Tabelle 9 anschaulich dargestellt sind. Hierbei stellt die Tabelle wiederum einen Auszug des Bewertungsbogens dar.

Nr.	Angaben zum Produkt und Anbieter	K.O. Kr.	GW Kr.	GW Gr.	Liquid XML Studio	XMetaL Author	Sydock	iReport	Jasper Assi-stant
2.1	Bekanntheitsgrad		15%	15%	8%	10%	5%	5%	5%
2.5	Referenzen		20%		15%	15%	0%	15%	0%
2.6	Benutzersupport		30%		20%	30%	25%	10%	10%
2.7	Add-ons/Plugins		10%		0%	0%	0%	0%	0%
2.8	Preis/Lizenz		25%		15%	5%	20%	25%	20%
<b>Summe</b>			<b>100%</b>		<b>58%</b>	<b>60%</b>	<b>50%</b>	<b>55%</b>	<b>35%</b>
<b>Gesamtgewicht in Gruppe</b>					<b>8,70%</b>	<b>9%</b>	<b>7,50%</b>	<b>8,25%</b>	<b>5,25%</b>
Nr.	Allgemeine Systemeigenschaften	K.O. Kr.	GW Kr.	GW Gr.	Liquid XML Studio	XMetaL Author	Sydock	iReport	Jasper Assi-stant
3.1	Unterstützt Windows Betriebssysteme	X	70%	10%	70%	0%	0%	0%	70%
3.4	Unterstützt JasperReports		30%		0%	0%	0%	30%	30%
<b>Summe</b>			<b>100%</b>		<b>70%</b>	<b>0%</b>	<b>0%</b>	<b>30%</b>	<b>100%</b>
<b>Gesamtgewicht in Gruppe</b>					<b>7%</b>	<b>0%</b>	<b>0%</b>	<b>3%</b>	<b>10%</b>
<b>Summe 1. und 2. Gruppe</b>					<b>15,70%</b>	<b>9,00%</b>	<b>7,50%</b>	<b>11,25%</b>	<b>15,25%</b>
<b>Rangfolge</b>					<b>4.</b>	<b>7.</b>	<b>8.</b>	<b>6.</b>	<b>5.</b>

(GW-Gewicht; Kr. - Kriterium; Gr. - Gruppe)

**Tabelle 9: Bewertung der Anbieter und Systemeigenschaften für die aussortierten Editoren**

Die Editoren Liquid XML Studio und JasperAssistant scheiterten an der vorgegebenen Auswahlgrenze. Außerdem wurden die XML-Editoren: XMetaL Autohor, Sydock und iReport weiteren Betrachtungen entzogen. Sie erfüllten nicht das vorgegebene K.O.-Kriterium „Unterstützung Windows Betriebssysteme“.

#### Angaben zum Produkt und Anbieter

Die aufgeführten Editoren wurden ebenfalls in den Gruppen „Angaben zum Produkt und Anbieter“ sowie „Allgemeine Systemeigenschaften“ bewertet. Nach Ansicht des Autors wurde ein Teil der Softwareprodukte aus Tabelle 9 bezüglich des Bekanntheits-

grades mit einem geringeren Prozentwert bewertet. Die Differenz unter den Anwendungen ging aus den Spezialisierungen dieser hervor. Die Produkte iReport und JasperAssistant sind in ihren Funktionen auf das ReportingTool JasperReports zugeschnitten, wodurch das Anwendungsgebiet dieser Produkte hinsichtlich Erstellung von XML-Dokumenten eingeschränkt ist. Größtenteils werden sie auf Websites und in Suchmaschinen nur in Verbindung mit JasperReports aufgeführt. Die übrigen WYSIWYG-fähigen Editoren unterstützen hingegen diverse DTDs und XML-Schemas sowie Stylesheets. Dies führt dazu, dass sie vielfältig einsetzbar sind und größtenteils benutzerdefinierte Anpassungen ermöglichen.

Nähere Untersuchungen der XML-Editoren iReport und JasperAssistant haben außerdem ergeben, dass offensichtlich keine Möglichkeit besteht, die vorhandene JasperReport DTD durch eine andere DTD zu ersetzen. Für den Einsatz der WYSIWYG-fähigen XML-Editoren in der SAB wäre diese Anpassung allerdings erforderlich.

Im Hinblick auf die Bewertungskriterien „Referenzen“ und „Benutzersupport“ erhielten alle Softwareprodukte eine hohe Gewichtung.

Aufgrund dessen, dass durch die Anbieter keine Add-ons für ihre Editoren zur Verfügung stehen, wurde das dazugehörige Kriterium jeweils mit 0 % bewertet.

Das Bewertungskriterium „Preis“ wurde überwiegend erhöht bewertet, da die WYSIWYG-fähigen Editoren zu angemessenen Preisen bzw. kostenlos angeboten werden.

## **Allgemeine Systemeigenschaften**

Den gravierenden Anlass zum Ausscheiden der Softwareprodukte XMetal Author, Sydock und iReport gab das K.O.-Kriterium „Unterstützung der Windows Betriebssysteme“ in der Gruppe „Allgemeine Systemeigenschaften“. Dadurch erfolgte keine weitere Betrachtung dieser Produkte.

Gemäß den Angaben der Anbieter wird das Betriebssystem Windows 2000 nicht unterstützt. Jedoch wird dieses derzeit in der SAB verwendet. Sollte es möglich sein, die betreffenden Editoren auf dem genannten Betriebssystem auszuführen, so ist dennoch der Benutzersupport durch den Anbieter nicht gewährleistet, der z.B. in Störungsfällen notwendig ist.

### **3.3 Einsatzbewertung der ausgewählten XML-Editoren**

Die ausgewählten Editoren werden nachfolgend hinsichtlich der praktischen Nutzung und bereitgestellten Funktionen näher untersucht und bewertet.

#### **3.3.1 Installation und Test**

Die fortführende Bewertung der spezifischen Funktionen an den ausgewählten Editoren setzt die Installation dieser voraus. Zur Evaluation der Softwareprodukte werden durch die Anbieter Testversionen bereitgestellt, die vollen Funktionsumfang bieten, jedoch in ihrer Nutzungsdauer eingeschränkt sind.

Der reelle Softwaretest geschah auf einem Einzelplatzrechner der SAB. Im Rahmen des Tests erfolgte entsprechend den Bewertungskriterien der Gruppen 4-8 eine Untersuchung der Oberfläche und Menüs der XML-Editoren. Dabei waren die verfügbaren Bearbeitungsansichten sowie Funktionen der Menü- und Symbolleiste von hoher Bedeutung. Weiterhin erfolgten zur Bewertung der Produkte Eingaben von Text, ProSAB-Keys und Grafiken.

#### **3.3.2 Detailbewertung anhand der Bewertungskriterien**

##### **Benutzeroberfläche**

Die Untersuchung der Benutzeroberfläche (siehe Tabelle 10) der ausgewählten XML-Editoren ergab, dass annähernd jedes Softwareprodukt über eine Textlayout- sowie WYSIWYG-Layout-Ansicht verfügt. Lediglich XMLmind enthält keine Textlayout-Ansicht. Somit bekam XMLmind nur für die WYSIWYG-Layout-Ansicht 30%. Die übrigen Editoren erhielten für beide K.O.-Kriterien jeweils einen Wert von 30%.

Die Softwareprodukte XMLSpy und Oxygen enthalten zusätzlich die Gridlayout-Ansicht, wodurch beide eine Bewertung von 15% erhielten.

Die Menüsprache sowie der Dialog zwischen Editor und Anwender finden annähernd bei allen Softwareprodukten in deutscher Sprache statt. Serna weist allerdings hinsichtlich der deutschen Übersetzung Mängel auf. So ist die Menüführung in englischer Sprache. Der Dialog durch Hinweismeldungen und Einstellungsfenster geschieht in deutscher Sprache, jedoch sind dabei Rechtschreibfehler und falscher Wortlaut festzustellen. Daher wurde eine Bewertung mit 7% vorgenommen.

Die Funktion der Seitenvorschau bieten wiederum nur XMLSpy und Oxygen, daher erhielten sie gegenüber den anderen zwei XML-Editoren eine Bewertung von 10%. XMLSpy enthält dafür eine Browser-Ansicht, die mittels eines CSS-Stylesheets den Inhalt des XML-Templates ausgibt. Oxygen ruft dagegen eine Standard Applikation (z.B. den Internet Explorer) auf, die Verwendung eines CSS-Stylesheets ist dabei ebenfalls unerlässlich.

Die Kumulierung der Prozentwerte ergab, dass die Softwareprodukte Oxygen und XMLSpy in der Gruppe „Benutzeroberfläche“ das beste Resultat erzielten. XMLmind wurde aufgrund der Nichterfüllung des K.O.-Kriteriums Textlayout-Ansicht aus weiteren Bewertungen ausgeschlossen. Es folgt die Bewertung der Gruppe „Entwurf, Dateneingabe und -bearbeitung“.

Nr.	Benutzeroberfläche	K.O. Kr.	GW Kr.	GW Gr.	XMLmind	Serna	XMLSpy 2009	Oxygen XML Editor
4.1	Textlayout-Ansicht	X	30%	20%	0%	30%	30%	30%
4.2	Gridlayout-Ansicht		15%		0%	0%	15%	15%
4.3	WYSIWYG-Layout-Ansicht	X	30%		30%	30%	30%	30%
4.4	Menüsprache - deutsch (Priorität) - englisch		15%		10%	7%	15%	15%
4.5	Seitenvorschau		10%		0%	0%	10%	10%
<b>Summe</b>			<b>100%</b>		<b>40%</b>	<b>67%</b>	<b>100%</b>	<b>100%</b>
<b>Gesamtgewicht in Gruppe</b>					<b>8,00%</b>	<b>13,40%</b>	<b>20,00%</b>	<b>20,00%</b>

(GW-Gewicht; Kr. - Kriterium; Gr. - Gruppe)

**Tabelle 10: Bewertung der Benutzeroberfläche**

## Entwurf, Dateneingabe und -bearbeitung

Die Erstellung von XML-Dokumenten unter der Verwendung von DTD und XSD wird durch die übrigen XML-Editoren unterstützt, wie der Tabelle 11 zu entnehmen ist. Daher wurde für das K.O.-Kriterium jeweils eine Bewertung von 20% vorgenommen.

Die Bereitstellung von kontextsensitiven Hilfen (Bewertungskriterium 5.2 und 5.3) werden in XMLSpy und Oxygen am besten realisiert. Sie bieten sowohl in der Textlayout-Ansicht, als auch WYSIWYG-Layout-Ansicht zahlreiche Möglichkeiten zum Einfügen und Ändern von Elementen. Die WYSIWYG-Layout-Ansicht des XMLSpy<sup>36</sup> ermöglicht zunächst Eingaben, die im SPS-Stylesheet vorgesehen sind, mittels der rech-

<sup>36</sup> In Altova Editoren entspricht die WYSIWYG-Layout-Ansicht der Authentic-Ansicht

ten Maustaste besteht allerdings die Möglichkeit, das Kontextmenü zu öffnen und weitere Elemente hinzuzufügen.

Oxygen<sup>37</sup> nutzt zur Darstellung von Inhalten innerhalb der WYSIWYG-Layout-Ansicht CSS-Stylesheets, diese dienen lediglich dem Layout der Ansicht. Somit stehen dem Entwickler weiterhin alle Funktionen zur Bearbeitung des XML-Templates zur Verfügung. Die Textlayout-Ansicht bietet hingegen in XMLSpy und Oxygen zahlreiche Funktionen, die das Erstellen von XML-Dokumenten sehr erleichtern. XMLSpy erhielt aufgrund der genannten Einschränkung jeweils 10% und Oxygen 15%. Serna bietet die Kontexthilfen lediglich in der WYSIWYG-Layout-Ansicht, daher wurden nur 7% vergeben.

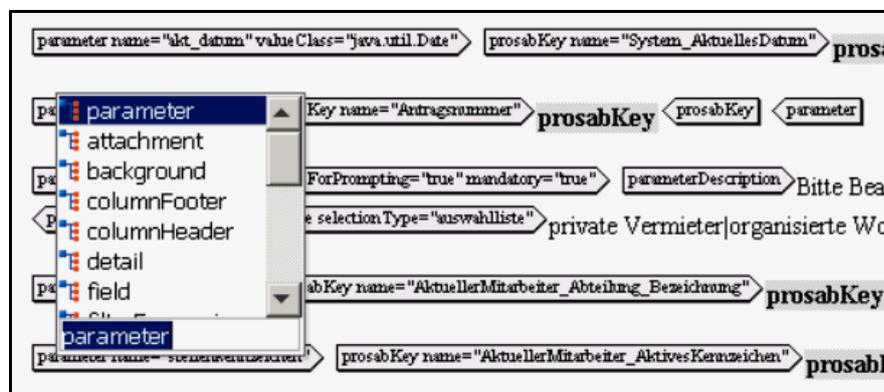


Abbildung 9: Kontextsensitive Eingabehilfe in Oxygen

Die Verwendung von Short Cuts und Funktionen, wie Suchen und Ersetzen, „Drag and Drop“ sind alltägliche Gebrauchsfunktionen eines Entwicklers. Sie erleichtern die Arbeit und bringen zudem eine Zeitersparnis. Für die Bereitstellung der Funktionen erhielt jeder XML-Editor zum Kriterium 5.5 eine Bewertung von 10%.

Das Kriterium 5.7 „Einfügen von Grafiken und Tabellen“ in Tabelle 11 ist ein wichtiger Bestandteil der XML-Templateerstellung. Dokumente enthalten des öfteren Logos im Dokumentkopf. Dazu bieten alle drei Editoren Eingabehilfen in der WYSIWYG-Layout-Ansicht, durch Bereitstellung eines Dialoges zum Öffnen eines vorhandenen Bilddokuments. Nach erfolgreicher Auswahl des Bilddokuments wird dieses an der vorgegebenen Stelle im XML-Template abgebildet sowie in der WYSIWYG-Layout-Ansicht dargestellt. Zum Einfügen von Tabellen stehen ebenfalls Funktionen in den Symbolleisten der XML-Editoren zur Verfügung. Jedoch sind diese Funktionen nur in Präsentationsformaten, wie DocBook bereitgestellt. Für den Einsatz persönlicher DTDs

<sup>37</sup> In Oxygen entspricht die Autor-Ansicht der WYSIWYG-Layout-Ansicht

bzw. der JasperReports DTD sind diese Funktionen in der WYSIWYG-Layout-Ansicht nicht vorgesehen. Aufgrund dessen, dass die vorhandenen Funktionen auf die ProSAB-DTD nicht anzuwenden sind, erhielten die Editoren jeweils eine Wertung von 0%.

Zur Bemessung des letzten Kriteriums 5.9 wurden in der WYSIWYG-Layout-Ansicht der Editoren, Elemente ohne und mit Inhalt eingefügt. Anschließend wurde der automatisch erstellte XML-Code in der Textlayout-Ansicht näher untersucht. Die Aufmerksamkeit galt hierbei dem Umfang des erstellten XML-Codes. Alle XML-Editoren erzeugten nur die eingegebenen bzw. laut DTD notwendigen Elemente und Attribute. Folgende Abbildung 10 stellt das Einfügen eines Elements ohne Inhalt deutlich dar, das eingefügte Element `<group>` kann in der Textlayout-Ansicht kurz bzw. ausführlich mit einem Start- und End-Tag dargestellt werden. In den untersuchten Editoren erfolgte stets die kurze Ausgabe. Daher bekamen alle Softwareprodukte für das Kriterium 5.9 eine Bewertung von 15%.

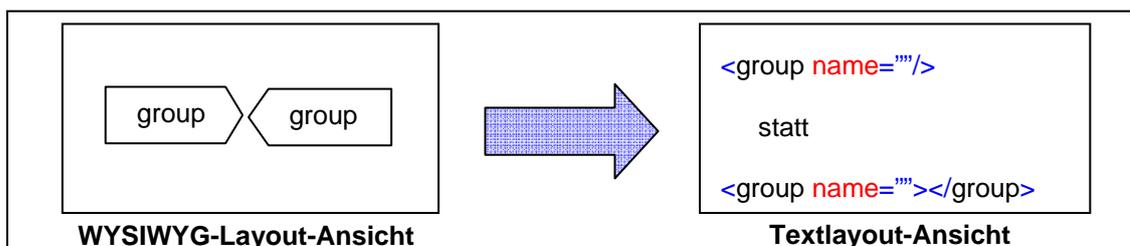


Abbildung 10: XML-Code-Darstellung

Die Zusammenfassung der vergebenen Bewertungsgewichte ergibt, dass Oxygen innerhalb dieser Gruppe das beste Ergebnis erzielt.

Nr.	Entwurf, Dateneingabe und -bearbeitung	K.O. Kr.	GW Kr.	GW Gr.	Serna Enterprise	XMLSpy 2009	Oxygen XML Editor
5.1	Unterstützung von DTD und XSD	X	20%	15%	20%	20%	20%
5.2	kontextsensitive Eingabemögl.		15%		7%	10%	15%
5.3	kontextsensitive Bearbeitungshilfen		15%		7%	10%	15%
5.5	Standardfunktionen und Short Cuts		10%		10%	10%	10%
5.7	Einfügen von Grafiken und Tabellen		15%		0%	0%	0%
5.9	Struktur, Umfang des generierten XML-Codes		15%		15%	15%	15%
<b>Summe</b>			<b>100%</b>			<b>59%</b>	<b>65%</b>
<b>Gesamtgewicht in Gruppe</b>					<b>8,85%</b>	<b>9,75%</b>	<b>11,25%</b>

(GW-Gewicht; Kr. - Kriterium; Gr. - Gruppe)

Tabelle 11: Bewertung der Dateneingabe

## Prüfroutinen

In Tabelle 12 erfolgt die Bewertung der Prüfroutinen. Ein K.O.-Kriterium stellt dabei die Validierung von XML-Templates dar. Wird ein Element bzw. Attribut, welches nicht der DTD bzw. XSD entspricht, innerhalb der XML-Editoren eingefügt, erscheint ein Hinweis für den Entwickler. Für die Einhaltung des K.O.-Kriteriums wurden jeweils 30% an die Softwareprodukte vergeben.

Die Funktion der „On the fly“-Validierung ist nur im Editor Oxygen verfügbar. Dabei werden die Eingaben des Entwicklers durch das Programm kontrolliert. Sofern Verstöße auftreten, wird der Nutzer umgehend darauf hingewiesen. Hierfür erhielt Oxygen eine Bewertung des Kriteriums mit 15%.

Die Wohlgeformtheit eines XML-Dokuments ist von hoher Bedeutung. Dem dazugehörigen K.O.-Kriterium 6.4 werden alle Softwareprodukte gerecht, daher wurden jeweils 25% erteilt.

Eine weitere nützliche Unterstützung ist die farbliche Kennzeichnung von fehlerbehaftetem XML-Code. Diese Funktion wird nur durch Oxygen bereitgestellt, in der Textlayout- und WYSIWYG-Layout-Ansicht werden Elemente und ihre Anordnung gegen die DTD validiert. Bei Verstößen wird der betroffene XML-Code rot unterstrichen, hierdurch wird der Entwickler auf die Fehler aufmerksam gemacht.

Nr.	Prüfroutinen	K.O. Kr.	GW Kr.	GW Gr.	Serna Enterprise	XMLSpy 2009	Oxygen XML Editor
6.1	Validierung des XML-Templates	X	30%	20%	15%	30%	30%
6.2	„On the fly“ Validierung		15%		0%	15%	15%
6.4	Prüfung auf Wohlgeformtheit	X	25%		25%	25%	25%
6.5	Hervorhebung von Syntaxfehlern in der Textlayout-Ansicht		20%		0%	0%	20%
6.6	Rechtschreibprüfung		10%		10%	10%	10%
<b>Summe</b>			<b>100%</b>		<b>50%</b>	<b>65%</b>	<b>100%</b>
<b>Gesamtgewicht in Gruppe</b>					<b>10,00%</b>	<b>13,00%</b>	<b>20,00%</b>

(GW-Gewicht; Kr. - Kriterium; Gr. - Gruppe)

**Tabelle 12: Bewertung der Prüfroutinen**

## Softwarequalität

Die Bewertung der XML-Editoren hinsichtlich der Softwarequalität erfolgt in Tabelle 13.

Nr.	Softwarequalität	K.O. Kr.	GW Kr.	GW Gr.	Serna Enterprise	XMLSpy 2009	Oxygen XML Editor
7.1	Funktionalität		30%	10%	8%	15%	20%
7.2	Zuverlässigkeit		15%		8%	15%	15%
7.3	Benutzbarkeit		25%		15%	15%	15%
7.4	Effizienz		10%		10%	7%	7%
7.5	Änderbarkeit		10%		10%	10%	10%
7.6	Übertragbarkeit		10%		10%	15%	15%
<b>Summe</b>			<b>100%</b>			<b>61%</b>	<b>77%</b>
<b>Gesamtgewicht in Gruppe</b>					<b>6,10%</b>	<b>7,70%</b>	<b>8,20%</b>

(GW-Gewicht; Kr. - Kriterium; Gr. - Gruppe)

**Tabelle 13: Bewertung der Softwarequalität**

In diesem Kontext stellt die Funktionalität der Softwareprodukte ein entscheidendes Bewertungskriterium dar. Letztendlich werden hohe Erwartungen an die Editoren hinsichtlich der WYSIWYG-Layout-Ansicht und weiterer Funktionen gestellt. Die Editoren XMLSpy und Oxygen bieten in der Textlayout-Ansicht vollständigen Funktionsumfang. Allerdings wurden erste Einschränkungen im Umgang mit der WYSIWYG-Layout-Ansicht festgestellt. In der Arbeit wurde bereits erwähnt, dass XMLSpy in der WYSIWYG-Layout-Ansicht ein SPS-Stylesheet voraussetzt. Dies führt dazu, dass im SPS-Stylesheet die zu verwendenden Elemente und Attribute, deren Anordnung im XML-Template sowie die Eingabefelder vorgegeben sind. Dadurch wird dem Entwickler zum Teil die Möglichkeit genommen, innerhalb der WYSIWYG-Layout-Ansicht die Form und den Inhalt des XML-Templates zu definieren. Das SPS-Stylesheet übernimmt hierbei die Rolle eines Templates. Die WYSIWYG-Layout-Ansicht gleicht einer Maske, die Eingabefelder zum Ausfüllen bereitstellt, wie nachfolgend in Abbildung 11 dargestellt.

Das CSS-Stylesheet gibt dagegen das Ausgabelayout des Elementinhalts vor. Dazu wird jedes Element der DTD im Stylesheet deklariert, zusätzlich erfolgt z.B. die Angabe der Schriftart, Formatierung, Schriftgröße, Farbe und Ausrichtung. In Abbildung 12 wird die WYSIWYG-Layout-Ansicht in Oxygen mit eingblendeten Tag-Symbolen dargestellt.



**Abbildung 11:**  
Ausschnitt WYSIWYG-  
Layout-Ansicht in  
XMLSpy 2009



**Abbildung 12:**  
Ausschnitt WYSIWYG-  
Layout-Ansicht in Oxygen



**Abbildung 13:**  
Ausschnitt WYSIWYG-  
Layout-Ansicht in Serna

Die Darstellung von Tags erfolgt in Serna ähnlich zum Editor Oxygen (siehe Abbildung 13). Das Anzeigen von Tag-Symbolen kann in allen drei Editoren ein- und ausgeschaltet werden. Aufgrund der unvollständigen Funktionen bezüglich des JasperReports und der mangelhaften Textlayout-Ansicht erhielt Serna eine Bewertung zum Kriterium 7.1 von 8%, XMLSpy 15% aufgrund des SPS-Stylesheets und Oxygen 20%.

Das nächste Bewertungskriterium ist die Zuverlässigkeit eines XML-Editors, dazu zählen z.B. die Submerkmale Reife, Fehlertoleranz und Wiederherstellbarkeit. Bezüglich dieser Merkmale ist Serna mangelhaft, der Dialog zwischen Anwendung und Nutzer enthält in deutscher Sprache Rechtschreibfehler. Weiterhin führt die falsche Anordnung eines Elementes im XML-Template dazu, dass aus der Textlayout-Ansicht der Wechsel in die WYSIWYG-Layout-Ansicht verhindert wird. Da die Textlayout-Ansicht keine Validierung und Fehleranzeige enthält, sind Fehler selbstständig und ohne Unterstützung zu beseitigen. Eine Validierung des XML-Templates erfolgt erst beim Öffnen der WYSIWYG-Layout-Ansicht. Aufgrund der aufgeführten Mängel wurde Serna eine Bewertung von 8% zugesprochen.

XMLSpy und Oxygen sind in dieser Hinsicht tolerant, sie ermöglichen trotz fehlerhaftem XML-Code die Anzeige der Bearbeitungsansichten. Außerdem werden die betreffenden Zeilen markiert bzw. in einem separaten Fenster aufgeführt und durch Hinweise zur Art des Fehlers ergänzt. Daher wurden beide Softwareprodukte mit 15% bewertet.

Es folgt das Kriterium 7.3, die Benutzbarkeit der einzelnen Editoren. Die Aufmerksamkeit wurde dabei unter anderem der Verständlichkeit, Erlernbarkeit und Bedienbarkeit gewidmet. Die drei vorliegenden XML-Editoren ermöglichen einen leichten Einstieg in die Nutzung. Die wichtigsten Funktionen sind in Symbol- und Menüleiste schnell ersichtlich. Insgesamt wurde jeweils eine Bewertung von 15% erteilt.

---

Ein weiteres Bewertungskriterium bildet die Effizienz der Editoren. Unter den gegebenen Bedingungen ist Serna problemlos einsetzbar, dies ist auf den geringeren Funktionsumfang zurückzuführen. Die Softwareprodukte XMLSpy und Oxygen beanspruchen dagegen mehr Rechenleistung, dies wird während der Ausführung offensichtlich. Bei näherer Betrachtung der laufenden Systemprozesse ist zu erkennen, dass die zwei Anwendungen häufiger den Prozessor beanspruchen. Während der Programmausführung werden zwischenzeitlich einzelne Eingaben verzögert aufgenommen und durchgeführt. Daher wurden Serna in der Bewertung 10% zugeteilt und den anderen beiden jeweils 7%.

Die Änderbarkeit stellt das nächste Bewertungskriterium dar. Diesbezüglich ermöglichen die XML-Editoren die Installation von Add-ons, Plugins sowie die Aktualisierung durch Updates. Die Bereitstellung dieser Module durch den Anbieter, ermöglicht eine fachliche und technische Anpassung der Softwareprodukte an Veränderungen. Neben einfachen Add-ons für Rechtschreibung und Menüführung in verschiedenen Sprachen, werden für Entwickler Plugins bereitgestellt, die eine Anbindung zusätzlicher Entwicklungsumgebungen ermöglichen. Daher wurde eine Bewertung von 10% pro Editor vergeben.

Zuletzt wurde die Übertragbarkeit der Softwareprodukte bewertet, hierbei waren die Anpassbarkeit der Editoren an andere Umgebungsbedingungen, Installierbarkeit und Koexistenz mit anderer Software von Bedeutung. Im Rahmen der Untersuchung wurden die XML-Editoren auf den Betriebssystemen Windows 2000 und XP erfolgreich ausgeführt. Außerdem werden durch die Anbieter weitere Betriebssysteme unterstützt. Zur Installierbarkeit ist anzumerken, dass bei der Installationen von Serna Komplikationen auftraten, die Auswirkungen auf einige Funktionen des Editors zur Folge hatten. Dementsprechend erhielt Serna 10% in der Bewertung, XMLSpy und Oxygen bekamen 15% zugeteilt.

## Softwareergonomie

In der letzten Gruppe erfolgt die Bewertung der Softwareergonomie (siehe Tabelle 14).

Nr.	Softwareergonomie	K.O. Kr.	GW Kr.	GW Gr.	Serna Enterprise	XMLSpy 2009	Oxygen XML Editor
8.1	Aufgabenangemessenheit		15%	10%	5%	10%	15%
8.2	Selbstbeschreibungsfähigkeit		20%		17%	15%	15%
8.3	Steuerbarkeit		20%		20%	20%	20%
8.4	Erwartungskonformität		15%		15%	15%	15%
8.5	Fehlertoleranz/ Fehlerrobustheit		10%		5%	9%	10%
8.6	Individualisierbarkeit		10%		5%	10%	10%
8.7	Lernförderlichkeit		10%		10%	10%	10%
<b>Summe</b>			<b>100%</b>		<b>77%</b>	<b>87%</b>	<b>95%</b>
<b>Gesamtgewicht in Gruppe</b>					<b>7,70%</b>	<b>8,70%</b>	<b>9,50%</b>

(GW-Gewicht; Kr. - Kriterium; Gr. - Gruppe)

**Tabelle 14: Bewertung der Softwareergonomie**

Das erste Bewertungskriterium bildet die Aufgabenangemessenheit. Sie besagt, dass Funktionalität und Dialog anhand der charakteristischen Eigenschaften der Arbeitsaufgabe gestaltet werden. Weiterhin ist die Arbeit des Entwicklers durch den XML-Editor zu vereinfachen. Dieser Anforderung entsprechen die Softwareprodukte XMLSpy und Oxygen. Neben der WYSIWYG-Layout-Ansicht bieten sie kontextsensitive Eingabehilfen und weitere Bearbeitungsfenster mit kontextbezogener Anzeige von Elementen und Attributen. Der XML-Editor Serna bietet ebenfalls diese Funktionen, allerdings sind sie geringer ausgeprägt. Eine Besonderheit weist Oxygen auf, die kontextsensitive Eingabehilfe führt zuerst Elemente und Attribute auf, die zuletzt eingefügt wurden. Diese Hilfsleistung erspart dem Entwickler das Scrollen durch die Liste bzw. die Eingabe der ersten Buchstaben des Elementes, damit es angezeigt wird. Daraus ergibt sich folgende Bewertung zum Kriterium 8.1, Serna erhielt 5%, XMLSpy 10% und Oxygen bekamen aufgrund der umfangreichen Funktionen 15%.

Es folgt das Bewertungskriterium 8.2 Selbstbeschreibungsfähigkeit mittels dessen die Verständlichkeit der Bedienung des Softwareprodukts bewertet. In diesem Zusammenhang ist festzustellen, dass die vorliegenden XML-Editoren über Tool-Tips und ausreichend beschriftete Dialoge verfügen. Der Nutzer ist in der Lage, die Funktionen der Editoren selbstständig zu nutzen, für aufkommende Fragen werden außerdem Benutzerhandbücher bereitgestellt. Diese sollten jedoch nur bei komplexen Problemen aufgeru-

fen werden, da andernfalls die Selbstbeschreibungsfähigkeit der XML-Editoren in Frage zu stellen ist. Aufgrund des Funktionsumfangs der Tools XMLSpy und Oxygen treten zeitweise Irritationen bezüglich der Einstellmöglichkeiten bzw. den Optionen auf, daher erhielten sie jeweils 15% in der Bewertung und Serna 17%.

Die Steuerbarkeit stellt ein weiteres Kriterium der Softwareergonomie dar, mit ihr wird die Steuerung der Dialoge zwischen System und Mensch beurteilt. Dabei wird der systemgesteuerte, benutzergesteuerte und gemischte Dialog unterschieden<sup>38</sup>. Die zu untersuchenden Softwareprodukte verwenden den benutzergesteuerten Dialog. Der Anwender veranlasst Aktionen, die nach ihrer Durchführung Ergebnisse zurückmelden, hierbei kann das System den Vorgang nicht beeinflussen<sup>39</sup>. Dieser Sachverhalt ist bei den vorliegenden XML-Editoren gegeben, daher wurden jeweilig 20% erteilt.

Das Bewertungskriterium 8.4 Erwartungskonformität ist ein weiteres Qualitätsmerkmal. Es ist zu beurteilen, inwiefern die Erwartungen der Nutzer durch das System erfüllt werden. Hierbei ist von Bedeutung, dass die Dialoge aus dem Nutzungskontext heraus vorhersehbaren Benutzerbelangen und allgemein anerkannten Konventionen entsprechen. Diesbezüglich erfüllen alle drei Softwareprodukte die Konventionen, Bezeichnungen in der Menüleiste sowie in den Dialogen sind standardgemäß. Demzufolge betrug die Bewertung jeweils 15%.

Nachfolgend wurden die XML-Editoren hinsichtlich des Bewertungskriteriums 8.5 Fehlertoleranz bewertet. Dabei war die Fehlerrobustheit des Systems von Bedeutung, das heißt das beabsichtigte Arbeitsergebnis sollte trotz fehlerhafter Eingaben des Nutzers erreicht werden. Dies beinhaltet das Beheben aufgetretener Fehler mit Unterstützung des XML-Editors. Die Untersuchung der Editoren ergab, dass in den Layout-Ansichten nur Eingabefehler abgefangen wurden, die gegen die Definition der DTD sprachen.

Innerhalb der Textlayout-Ansicht besteht ein erhöhtes Risiko, dass Fehler hinsichtlich Wohlgeformtheit und Validität des XML-Templates auftreten, da der Anwender beliebig Zeichen im XML-Code ändern kann. Daher sind Funktionen, wie „On the fly“ Validierung enorm wichtig, diese verweisen rechtzeitig auf Fehler. Ebenso sind die bereits in Gruppe 6 genannten Prüfroutinen entscheidend, eine farbliche Kennzeichnung der betroffenen Zeilen bzw. Aufzählung dieser in einem zusätzlichen Fenster erleichtert dem Nutzer die Fehlersuche. Oxygen erhielt aufgrund der vorhandenen Prüfroutinen und der „On the fly“-Validierung 10%, XMLSpy verfügt leider nicht über die Echtzeitüberprüfung, daher wurden 9% vergeben. Das Softwareprodukt Serna verfügt ebenfalls

---

<sup>38</sup> vgl. HERCZEG, M. (2009), S. 180

<sup>39</sup> vgl. Ebenda

---

über die Prüfroutinen, enthält aber keine „On the fly“-Validierung und gibt keine Unterstützung in der Textlayout-Ansicht, daher betrug die Bewertung nur 5%.

Das nächste Bewertungskriterium lautet Individualisierbarkeit. Damit die Nutzung eines Softwareprodukts den Nutzerbedürfnissen entspricht, sind Funktionen zur Gestaltung der Dialoge und Bearbeitungsansicht bereitzustellen. Dazu bieten die vorhandenen XML-Editoren die Möglichkeit, die Symbolleisten an eigene Ansprüche anzupassen. Für den Aufruf von Programmfunktionen können durch den Nutzer die entsprechenden Symbole angeordnet bzw. ausgeblendet werden. Weitere Einstellungen ermöglichen XMLSpy und Oxygen durch Anpassung der Bearbeitungsfenster, indem die Fenster ein- und ausgeblendet werden. Daraus ergibt sich folgende Bewertung, Serna erhielt 5%, XMLSpy und Oxygen erhielten jeweils 10%.

Zuletzt sind die XML-Editoren hinsichtlich des Kriteriums 8.7 Lernförderlichkeit zu bewerten. Mit der Lernförderlichkeit werden Dialoge, die den Nutzer beim Kennenlernen des Editors unterstützen und anleiten, beurteilt. Diesbezüglich sind Hilfestellungen durch das Softwareprodukt bereitzustellen. Folgende Hilfestellung sind vorstellbar: das Anzeigen von Tool-Tips, Benutzerhandbuch und die Bereitstellung von Informationen, wenn zu einem Menüpunkt die Hilfefunktion aufgerufen wird. Weiterhin sind Statusmeldungen bei mehreren Arbeitsschritten bzw. am Ende einer ausgeführten Aktion einzublenden. In Bezug auf die genannten Funktionen weisen die untersuchten Softwareprodukte keine Schwächen auf, sodass je Editor 10% vergeben wurden.

### 3.3.3 Auswahl des XML-Editors

Die Auswahl des XML-Editors erfolgte auf Grundlage der Bewertungsgewichte im Abschnitt 3.3.2. Die Kumulierung der einzelnen Gruppengewichte je Softwareprodukt wird nachfolgend dargestellt (siehe Tabelle 15).

Nr.	Gruppenname	Gewicht der Gruppe	Serna Enterprise	XMLSpy 2009	Oxygen XML Editor
2	Angaben zum Produkt und Anbieter	15%	9,00%	12,00%	9,75%
3	Allgemeine Systemeigenschaften	10%	7,00%	7,00%	7,00%
4	Benutzeroberfläche	20%	13,40%	20,00%	20,00%
5	Entwurf, Dateneingabe und -bearbeitung	15%	8,85%	9,75%	11,25%
6	Prüfroutinen	20%	10,00%	13,00%	20,00%
7	Softwarequalität	10%	6,10%	7,70%	8,20%
8	Softwareergonomie	10%	7,70%	8,70%	9,50%
<b>Summe der Gr. GW</b>		<b>100%</b>	<b>62,05%</b>	<b>78,15%</b>	<b>85,70%</b>

(GW-Gewicht; Kr. - Kriterium; Gr. - Gruppe)

**Tabelle 15: Gesamtbewertung der ausgewählten Editoren**

An den kumulierten Gruppengewichten in Tabelle 15 ist zu erkennen, dass Oxygen das beste Resultat erzielt. An zweiter Stelle folgt das Softwareprodukt XMLSpy 2009. Den dritten Platz belegt Serna. Es entspricht am wenigsten den gestellten Anforderungen.

Den entscheidenden Unterschied zwischen Oxygen und XMLSpy stellt die WYSIWYG-Layout-Ansicht dar. Die Verwendung des SPS-Stylesheets im XMLSpy bzw. alternativ im kostenlosen Editor Authentic von Altova verschafft viele Vorteile. Das Layout und der Inhalt für das zu erstellende XML-Dokument werden im SPS-Stylesheet vorgegeben. Weiterhin können Daten aus relationalen Datenbanken hinzugefügt werden, die anschließend in das XML-Dokument einfließen. Jedoch ist festzustellen, dass das SPS-Stylesheet einem Template entspricht und für die Erzeugung von Berichten als nicht geeignet erscheint. Es ist möglich, ein SPS-Stylesheet als Eingabemaske zur Erstellung von XML-Templates zu verwenden (siehe Abbildung 14).

JasperReports					
name	leftMargin	bottomMargin	rightMargin		
entscheidungsvorlage_darlehen	70	55	52		

ReportFont					
name	isDefault	fontName	size	isBold	pdfFont
Arial_Bold	false ▾	Arial	11	true ▾	Helvetica-B
Arial_Small	false ▾	Arial	8	<a href="#">@isBold hinzufügen</a>	Helvetica

Abbildung 11: Entwurf einer Eingabemaske mit SPS-Stylesheet

Allerdings entspricht diese Ansicht nicht dem aufgestellten Soll-Konzept zur Erstellung von ProSAB-Templates. Daher wird nachfolgend im Abschnitt 4.1.3 ein ausführlicher Test mit dem XML-Editor Oxygen unternommen. Die Verwendung des CSS-Stylesheets zur Darstellung von XML-Templates in der WYSIWYG-Layout-Ansicht spricht dafür, da gegenüber dem SPS-Stylesheet keine Vorgaben hinsichtlich der Anordnung von Elementen vorliegen. Ein weiterer Vorteil besteht darin, dass zur Eingabe der Attributwerte in Oxygen ein zusätzliches Fenster zur Verfügung steht.

## 4 Prototypische Realisierung unter Anwendung von Oxygen

Der XML-Editor Oxygen erhielt in der Untersuchung von Softwareprodukten die höchste Bewertung. Er wurde daher einem umfangreicheren Test unterzogen, indem das ProSAB-Template EEV in der WYSIWYG-Layout-Ansicht erstellt wurde.

### 4.1 Was ist eine Entscheidungsvorlage

Die Kreditvergabe an natürliche<sup>40</sup> und juristische<sup>41</sup> Personen erfordert ausführliche Informationen zum Antragsteller. Anhand der Daten beurteilt der Sachbearbeiter die momentane und zukünftige Entwicklung des Kreditsuchenden. Hierbei sind unter anderem die Kreditfähigkeit, Kreditwürdigkeit, wirtschaftliche Verhältnisse und die Vermögenssituation zu prüfen.

Zur Beurteilung der Risiken, die mit einer Kreditvergabe verbunden sind, wird unter anderem eine Bonitätsanalyse durchgeführt. Diese enthält die Analyse der Kreditwürdigkeit und des Besicherungspotenzials.<sup>42</sup> Die Kreditwürdigkeit wird mittels der Jahresabschlussanalyse, Betriebsbesichtigungen und Kreditgesprächen ermittelt. Die Analyse des Besicherungspotenzials erfolgt durch Gutachten.

Die gewonnenen Informationen sowie positive und negative Merkmale werden in einer Entscheidungsvorlage aufgeführt. Sie bildet die Grundlage für Geschäftsvorfälle in einem Kreditinstitut. Anhand des Dokuments sind durch den Sachbearbeiter die möglichen Risiken des Kredits und die geschäftspolitischen Ziele der Institution gegeneinander abzuwägen. Außerdem ist ein Votum abzugeben. Im Anschluss ist aufgrund des Vier-Augen-Prinzips<sup>43</sup> die Entscheidungsvorlage dem Kompetenzträger zu übergeben. Dieser ist vom Kreditinstitut befugt, Kredite bis zu einer bestimmten Höhe zu bewilligen. Entsprechend der Kredithöhe kann eine Zustimmung durch das Aufsichtsgremium erforderlich sein.

Neben der Kompetenzordnung sind gesetzliche Grundlagen, vor allem das Kreditwesengesetz sowie Arbeitsanordnungen des Kreditinstituts zu beachten.

---

<sup>40</sup> rechtsfähiger Mensch

<sup>41</sup> rechtlich verselbstständigte Personenmehrheit (Organisation)

<sup>42</sup> vgl. SCHILLER B./TYTKO D. (2001), S. 76

<sup>43</sup> Entscheidungen sind durch 2 unabhängige Personen zu treffen

Entscheidungsvorlagen werden unter anderem für Neuengagement bzw. Engagementsweiterungen im Kreditgeschäft verwendet. Darüber hinaus dienen sie dem Folgegeschäft z.B. Sicherheitenänderungen, Sicherheitenfreigaben, Prolongation<sup>44</sup> und Schuldnerwechsel.

## **4.2 Dokumentaufbau der Entscheidungsvorlage für Darlehen**

In diesem Abschnitt werden die einzelnen Bestandteile der zu erstellenden Entscheidungsvorlage für Darlehen näher dargestellt. Hierzu befindet sich im Anhang (siehe Anlage 3) der vorliegenden Arbeit die Vorlage der EEV.

Auf der ersten Seite der EEV sind besondere Eigenschaften des Kredits (z.B. Normal-, Groß- oder Organkredit) zu vermerken. Weiterhin wird die Art des Darlehens (z.B. Neuausreichung) angegeben. Anschließend folgen allgemeine Kunden- und Branchenangaben. Hierbei entspricht der Antragsteller gleichzeitig dem Kreditnehmer. Neben dem Antragsteller können weitere Kreditnehmer aufgeführt werden. Diese bilden zusammen eine Kreditnehmereinheit. Jede Kreditnehmereinheit verfügt über eine Konzernnummer und Konzernbezeichnung, die in der Entscheidungsvorlage erscheinen.

Es folgt der erste Absatz der EEV, er stellt eine Engagementübersicht des Kreditnehmers bzw. der Kreditnehmereinheit dar. In der Übersicht sind z.B. aktuelle Risiken und Inanspruchnahme/Auszahlungsreste des Kreditnehmers bzw. Kreditnehmereinheit aufgelistet. Weiterhin werden beantragte Veränderungen und künftige Risikobeträge, die zum Beispiel durch ein Neugeschäft<sup>45</sup> entstehen, in der Engagementübersicht dargestellt.

Zusätzlich erfolgt eine Auflistung der bestehenden/zukünftigen Gläubiger der bereits ausgezahlten/noch auszahlenden Kredite.

Ein weiterer wichtiger Bestandteil des Absatzes sind Evidenzmeldungen der Kreditnehmer, sofern diese vorliegen. Kredit- und Finanzdienstleistungsinstitute sind vierteljährlich verpflichtet, vergebene Millionenkredite (ab 1,5 Mio. Euro Volumen) der Deutschen Bundesbank mitzuteilen.<sup>46</sup> Es besteht ebenfalls die Möglichkeit, dass Unternehmen auf Antrag den Schuldenstand eines Kreditnehmers bzw. einer Kreditnehmereinheit durch die Deutsche Bundesbank mitgeteilt bekommen.<sup>47</sup>

---

<sup>44</sup> Verlängerung der Zinsbindungsfrist für ein laufendes Darlehen unter Anpassung des Zinssatzes

<sup>45</sup> Kredite, die einem Kreditnehmer/Kreditnehmereinheit von der SAB gewährt werden

<sup>46</sup> vgl. § 14 Abs. 1 Satz 1 KWG, dazu BOOS, K.-H./FISCHER R./SCHULTE-MATTLER H. (2004)

<sup>47</sup> vgl. § 14 Abs. 2 Satz 4 KWG, dazu BOOS, K.-H./FISCHER R./SCHULTE-MATTLER H. (2004)

Den zweiten Absatz der EEV bilden die rechtlichen Verhältnisse. Sofern die Kreditnehmer ausschließlich aus juristischen Personen bestehen, sind diese einzeln aufzuführen. Dabei werden z.B. Informationen aufgelistet, wie. Name der Gesellschaft, Rechtsform und die Höhe der Kapitaleinlage.

Den nächsten Absatz bilden die Sicherheiten und Beleihungswerte. In tabellarischer Form werden die bestehenden sowie neu hereinzunehmenden Sicherheiten aufgeführt. Sicherheiten können z.B. Grundschulden und Bürgschaften des Freistaates Sachsen sein.

Es folgt der Absatz Verwendungszweck bzw. Vorhabensdarstellung. Anträge auf Neuzusage beinhalten eine Vorhabensdarstellung. Hierzu ist das geplante Vorhaben verbal darzustellen. Außerdem sind Kosten und Finanzierung der Neuinvestition aufzuführen. Neben einem Antrag auf Neuzusage kann eine Darlehensablösung (teil-) sanierter Objekte erfolgen. In diesem Fall ist der aktuelle Sanierungsgrad der Immobilie darzustellen.

Weiterhin kann eine Objektwirtschaftlichkeitsberechnung erfolgen. Innerhalb der Berechnung werden die Kosten und Erträge des Investitionsobjekts gegenübergestellt. Daraus ergibt sich der jährliche Überschuss bzw. Fehlbetrag des Objekts. Alternativ kann die Ermittlung einer Einzelwertberichtigung vorgenommen werden. Demnach findet eine Neubewertung von Forderungen unter Berücksichtigung von erkannten bzw. absehbaren Ausfallrisiken statt. In diesem Zusammenhang verfügt die SAB über ein Risikovorsorgeprognosesystem. Das System dient unter anderem der Überwachung von Engagements und frühzeitigen Identifizierung von Risiken. Treten Umstände ein, die noch nicht berücksichtigt wurden und die den Sicherheitenwert negativ beeinflussen, z.B. der Verfall von Bausubstanz oder Mietausfall, so wird im Risikomanagementsystem eine Rotmeldung ausgegeben. Sofern Rotmeldungen des Kreditnehmers bzw. der Kreditnehmereinheit vorliegen, werden sie in der EEV unterhalb der Einzelwertberichtigung aufgeführt.

Im nächsten Absatz werden die wirtschaftlichen Verhältnisse des Kreditnehmers aufgeführt. Hierzu ist die Notwendigkeit zur Offenlegung der wirtschaftlichen Verhältnisse nach § 18 KWG in der Entscheidungsvorlage zu dokumentieren. Weiterhin wird die Risikoklasse bzw. das Rating<sup>48</sup> des Kreditnehmers für das aktuelle und vorhergehende Jahr ausgewiesen. Für Kreditnehmer der organisierten Wohnungswirtschaft sind darüberhinaus die Planzahlen zur nachhaltigen Kapitaldienstfähigkeit anzugeben.

---

<sup>48</sup> Bonitätsanalyse, Bewertung des Kreditnehmers anhand der Basel II Richtlinien

Zusätzlich sind die Ergebnisse der Bonitätsanalyse durch den Sachbearbeiter zu werten. Weiterhin ist eine Aussage zur Risikovorsorge zu treffen.

Im vorletzten Absatz der Entscheidungsvorlage werden Bedingungen und Auflagen, die an das Einzelengagement gebunden sind, aufgeführt.

Den letzten Absatz der Entscheidungsvorlage bilden das Votum, die Zusammenfassung und die Engagementstrategie. In diesem Zusammenhang ist durch den Sachbearbeiter eine Gegenüberstellung der Chancen und Risiken vorzunehmen. Zusätzlich muss er aus den Abwägungen heraus einen eindeutigen Entscheidungsvorschlag an den Kompetenzträger ableiten.

### 4.3 ProSAB- und JasperReports-Bestandteile des Templates Entscheidungsvorlage

ProSAB-Templates verwenden Funktionen, wie Subreports und Keys, die durch JasperReports und ProSAB bereitgestellt werden und für die Umsetzung des Templates EEV relevant sind.

Nachfolgend wird ein Ausschnitt aus der Vorlage der EEV (siehe Abbildung 15) dargestellt. In diesem wurden die Bestandteile sowie deren Datenherkunft farblich gekennzeichnet. Eine Erläuterung der Markierungen erfolgt anhand einer Legende.

<b>Beschluss</b>	Dresden, den 26.10.2009
Der <b>Kompetenzträger</b> beschließt	
1.	
2.	
Es werden Beleihungswerte in Höhe von insgesamt € festgesetzt.	Datum, Unterschrift
letzter Beschluss: gemeint ist der letzte risikorelevante Beschluss	Kompetenzträger:
Gesamtengagement bei letztem Beschluss in T€	Charakter der Vorlage:
<b>Begründung für die vorgelegte Kompetenz:</b>	
<input checked="" type="checkbox"/> Normalkredit	
<input checked="" type="checkbox"/> Neuausreichung Darlehen	
<input checked="" type="checkbox"/> Erhöhung / Bildung / Auflösung einer Einzelwertberichtigung	

## 1. Kreditnehmer

Name:		Kunden-Nr.:	000-000000
Sitz:			
Gründungsdatum:			
Branche:		Konzern-Nr.:	0000000

Abbildung 12: Ausschnitt aus der Vorlage der EEV

### Legende:

1.  Festtext: Text ist definierter Bestandteil des ProSAB-Templates
2.  Baustein: Integration bestehender bzw. Neuanlage von Bausteinen, Daten werden zum Teil aus der Datenbank entnommen bzw. es erfolgen Benutzereingaben
3.  ProSAB: Die benötigten Daten werden mittels Listen und Keys durch ProSAB bereitgestellt
4.  Textfeld: Kann optional bzw. zwingend eine Texteingabe vom Sachbearbeiter erfordern
5.  Auswahlliste: Ein Feld, indem der Sachbearbeiter optional bzw. zwingend eine Auswahl trifft

Hervorgehobener Dokumenttext, dessen Farbton der Legenden-Ziffer eins entspricht, stellt Festtext dar. Dazu zählen vor allem Überschriften, Sätze und Bezeichnungen, die durch die Fachabteilung vorgegeben sind. Die Vorgaben werden vom Entwickler im Template eingefügt. Eine nachträgliche Änderung des Textes ist während der späteren Ausführung in ProSAB ausgeschlossen. Festtext wird durch die JasperReports Elemente `<textField>` und dessen Kindelement `<textFieldExpression>` im Template festgelegt.

Dokumentbestandteile, die gemäß Legenden-Ziffer zwei markiert sind, stellen Bausteine dar. Diese vorhandenen Templates können in die zu erstellende Entscheidungsvorlage eingebunden werden bzw. müssen zuvor neu angelegt werden. In diesem Zusammenhang ist z.B. der Baustein „Kreditnehmer“ neu zu erstellen. Bausteine dienen vorwiegend der Darstellung von Tabellen und vorbelegten Texten, die mehrfach verwendet werden. Für die Ausgabe von Daten innerhalb der Tabelle ist ein `<tableFilter>` zu vereinbaren. Dieser verwendet eine Liste des ProSAB-Objektmodells, die `<prosabList>`. Listen enthalten ein bzw. mehrere Objekte. Einzelne Werte eines Listenobjektes werden anhand von Detail-Abfragen mittels `<prosabKey>` ermittelt. Im Anschluss daran werden die Daten durch das Element `<field>` im Baustein bereitgestellt.

Hervorgehobener Text bzw. Zahlenfelder der Vorgabe, deren Farbton der Legenden-Ziffer drei entspricht, stellen Daten dar, die durch ProSAB bereitgestellt werden. Dies sind z.B. Angaben, wie zuständiger Mitarbeiter, Kundenangaben oder Zahlenwerte in Bausteinen. Die Ausgabe der Daten im Schriftstück erfolgt kontextabhängig von ProSAB. Sofern im Rahmen der Antragsbearbeitung in ProSAB mehrere Objekte zur Verfügung stehen, müssen *<prosabList>*-Elemente eventuell im Zusammenhang mit einem *<filter>* im ProSAB-Template angewendet werden. Im *<filter>* können anhand von Attributen Bedingungen aufgestellt werden, wodurch die Anzahl der Objekte einer Liste minimiert wird. Danach können mittels *<prosabKey>* die konkreten Werte der Objekte abgefragt werden. ProSAB-Keys und -Listen werden zu Beginn des Templates im XML-Code anhand von *<parameter>* Elementen vereinbart. Durch Aufruf des Parameternamens im Template kann der Rückgabewert des *<prosabKey>* im Schriftstück ausgegeben werden.

Das Symbol der Legenden-Ziffer vier kennzeichnet einzeilige und mehrzeilige Eingabefelder für Text. Sie werden ebenfalls zu Beginn des Templates durch das Element *<parameter>* angelegt. Mittels Attributen werden sie als Eingabe-, Pflicht- und Textfeld definiert. Die Positionierung des Eingabefeldes erfolgt wiederum durch Aufruf des Parameternamens im Element *<textField>* bzw. *<subreport>*. Die definierten Eingabefelder werden während der Ausführung des Templates in der ProSAB-Bearbeitungsansicht bereitgestellt. Texteingaben, die innerhalb der Ansicht durch den Sachbearbeiter erfolgen, werden bei der Erstellung des Schriftstückes mit einbezogen. Entsprechend den Angaben im XML-Code des Templates wird der Text im Schriftstück platziert. In Abbildung 15 entspricht das Eingabefeld zum Beschluss des Kompetenzträgers dem dargestellten Sachverhalt. Innerhalb der ProSAB-Bearbeitungsansicht ist der Sachbearbeiter verpflichtet, Texteingaben vorzunehmen. Diese mehrzeiligen Textfelder werden als Freitextfelder bezeichnet, da der Sachbearbeiter beliebige Eingaben und Formatierungen tätigen kann.

Das letzte Symbol der Legenden-Ziffer fünf markiert Auswahllisten. In diesem Zusammenhang werden Eingabefelder als Auswahllisten in der ProSAB-Bearbeitungsansicht bereitgestellt. Sie dienen der Steuerung des Templates indem konkrete Abfragen, so genannte *printWhenExpressions*, bezüglich des Parameterwertes des Eingabefeldes gestellt werden. In der Vorgabe dienen z.B. die Angaben (private oder organisierte Wohnungswirtschaft), die zu Beginn des Dokuments aufgeführt werden als Abfragewerte. Das Anlegen dieser Eingabefelder erfolgt ebenfalls durch das Element *<parameter>*.

Dabei wird allerdings das Kindelement `<defaultValue>` mit der Definition der Auswahlliste und dem Attribut `selectionType` zusätzlich vereinbart. Dadurch wird das Eingabefeld als Auswahlliste in der ProSAB-Bearbeitungsansicht dargestellt.

#### **4.4 Erstellen des Templates Entscheidungsvorlage mit dem ausgewählten XML-Editor**

Im Folgenden werden ausgewählte Inhalte des XML-Templates EEV dargestellt. Insbesondere wird die Verwendung von Templatebestandteilen, wie Auswahllisten, Bausteinen und Keys mit dem Editor Oxygen vorgestellt.

Für die Nutzung der WYSIWYG-Layout-Ansicht von Oxygen ist zunächst ein CSS-Stylesheet zu erstellen. Oxygen unterstützt XML-Formate wie DocBook. Hierfür sind alle Funktionen der Anwendung verfügbar. Eine unbekannte DTD z.B. von JasperReports kann in der WYSIWYG-Layout-Ansicht von Oxygen nicht dargestellt werden. Hierfür ist ein separates CSS-Stylesheet aufzurufen. Daher wurde im Rahmen der Arbeit dazu ein Stylesheet entworfen, das die Darstellung der einzelnen Elemente innerhalb der Ansicht definiert. Die fehlende Unterstützung von JasperReports durch Oxygen hat zur Folge, dass Gestaltungsfunktionen wie Formatierung des eingegebenen Textes in der WYSIWYG-Layout-Ansicht nicht zur Verfügung stehen. Dadurch wird die Erstellung von Templates teilweise erschwert bzw. die Funktionalität ist nicht vollständig gegeben. Daher wird nachfolgend die Erstellung des Templates EEV mit eingeschränkten Funktionen dargestellt.

In der vorliegenden Entscheidungsvorlage sind die Antragsteller zwischen privaten Vermietern<sup>49</sup> und der organisierten Wohnungswirtschaft<sup>50</sup> zu unterscheiden. Entsprechend der Zugehörigkeit des Antragstellers werden in den einzelnen Absätzen gezielt Textpassagen und Tabellen aufgeführt. Hierzu sind im Template Steuerungen in Form von Abfragen vorzunehmen.

Bei der Erstellung des Schriftstücks EEV wird in der ProSAB-Bearbeitungsansicht zunächst ein Auswahlfeld (siehe Abbildung 13) angezeigt, um eine Zuordnung des Antragstellers vorzunehmen. Dazu wird im Template ein entsprechender Parameter definiert (siehe Abbildung 14). Dieser Screenshot zeigt einen Ausschnitt der WYSIWYG-Layout-Ansicht.

---

<sup>49</sup> Personen, die in Besitz befindliche Wohnungen außergewerblich vermieten.

<sup>50</sup> Gewerbetreibende Unternehmen, die ausschließlich Wohnungen und Gebäude vermieten.

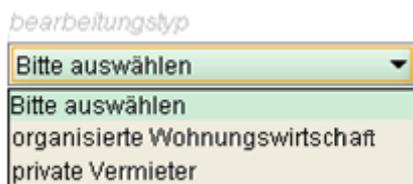


Abbildung 13: Auswahlliste in ProSAB-Bearbeitungsansicht

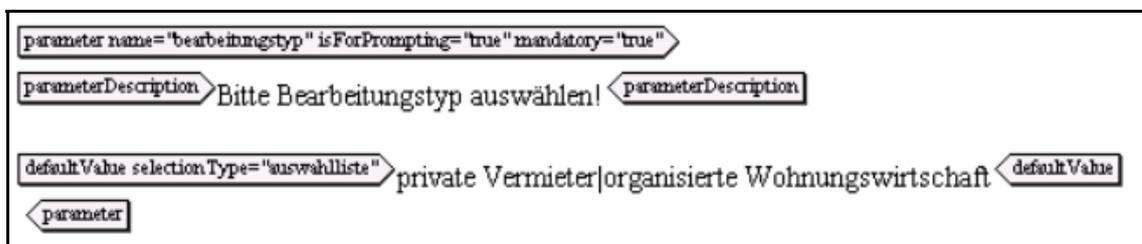


Abbildung 14: Einfügen eines Parameters in WYSIWYG-Layout-Ansicht von Oxygen

Es wurde bereits im letzten Abschnitt 4.3 erwähnt, dass die ProSAB-Auswahlliste durch einen `<parameter>` definiert wird. Das Einfügen eines Elements ist hierbei sehr benutzerfreundlich durchzuführen. Durch Drücken der Enter-Taste wird eine „Drop-Down“ Auswahlliste hervorgerufen. In dieser kann ein Element ausgewählt und durch Bestätigung eingefügt werden. Alternativ steht rechts neben dem Bearbeitungsfeld ein weiteres Fenster mit Elementen zur Verfügung. Ein doppelter Mausklick auf ein Element führt dazu, dass es im Bearbeitungsfenster eingefügt wird. Hierbei ist positiv hervorzuheben, dass die genannte Auswahlliste sowie das Fenster nur Elemente zum Einfügen aufführen bzw. hervorheben, die im aktuellen Bearbeitungskontext laut DTD erlaubt sind.

Die Eingabe von neuen Elementen erfordert zum Teil die Angabe von Attributen und des dazugehörigen Wertes. Hierzu bietet Oxygen ebenfalls ein separates Eingabefenster, das rechts vom Bearbeitungsfenster (siehe Abbildung 15) angeordnet ist.

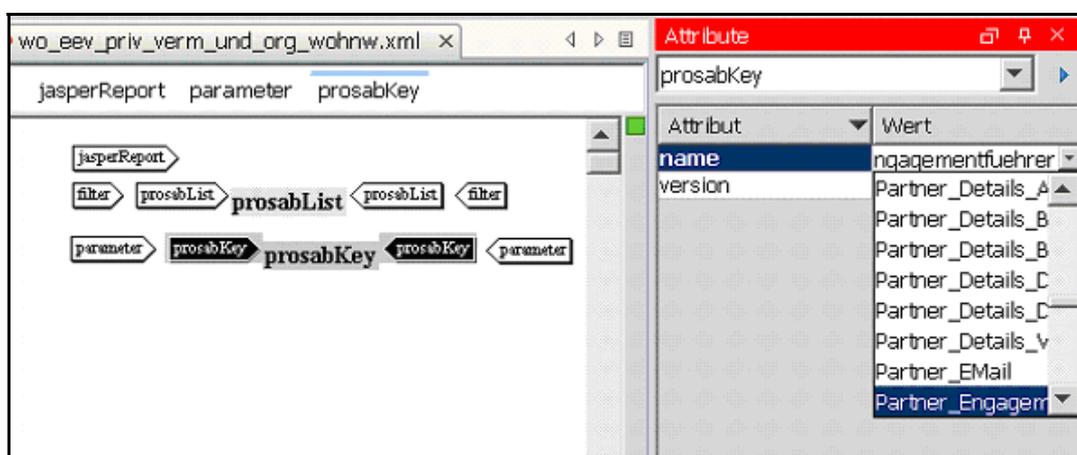


Abbildung 15: Auswahllisten für die Eingabe von Attributwerten

In Abhängigkeit vom ausgewählten Element des Bearbeitungsfensters werden im rechten Fenster die einzelnen Attribute des Elements und ihre möglichen Werte, die sie laut DTD einnehmen können, dargestellt.

Auf diese Art können ebenfalls ProSAB-Listen und -Keys eingefügt werden. Sie enthalten jeweils das Attribut *name*, womit ein bestimmter Objektwert in ProSAB abgefragt wird. Daher ist es sehr hilfreich, dass die zahlreichen, unter Umständen recht langen Namen, in einer Auswahlliste (siehe Abbildung 13) bereitgestellt werden. Nach der Wahl des Attributwertes wird dieser im XML-Code des Templates festgehalten. In Abbildung 15 sind diesbezüglich die Symbole der Elemente ausführlicher dargestellt. Die Attribute und ihre Werte werden dabei zusätzlich angezeigt.

Neben Eingabefeldern und Listen erfolgt das Einfügen von Bausteinen ebenfalls mittels Parameter. Hierbei wird das Kindelement `<parameterDescription>` verwendet, welches zur Angabe der Bausteinnummer dient. Damit wird das zu verwendende Template referenziert. Diesbezüglich stellt die Funktion des Elements eine ProSAB Besonderheit dar. Die genaue Position des Bausteininhalts wird durch das Element `<subreport>` im Template bestimmt. Dabei werden im Kindelement `<reportElement>` die Koordinaten des Textes bzw. der Tabelle mittels Attributen angegeben.

Weitere Eingabefelder, die häufig angewandt werden, sind Textfelder. Sie können als einzeilige und mehrzeilige Felder im Element `<parameter>` definiert werden. Zugleich kann im Kindelement `<defaultValue>` eine Textvorbelegung erfolgen. Sie wird innerhalb der ProSAB-Bearbeitungsansicht in Eingabefeldern angezeigt. Ein angelegter Parameter für ein Freitextfeld mit Textvorbelegung ist in Abbildung 16 dargestellt.

```

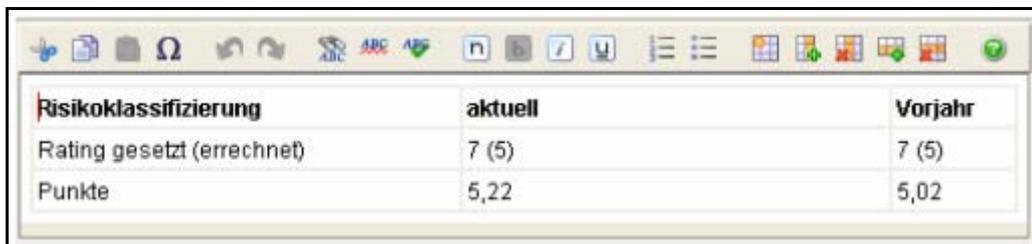
parameter name='beschreibung' class='de.sachsen.sab.it.prosab.bl.textverwaltung.ProSabJasperReport' isForPrompting='true' >
  <defaultValue >
    <p > Der
    Vorstand/Geschäftsbereichsleiter, Abteilungsleiter, Gruppenleiter beschlie&#223;t
    </p >
    <ol >
      <li >
      </li >
    </ol >
    <p > Es werden Beleihungswerte in H&#246;he von insgesamt
    &#364;, festgesetzt. </p >
  </defaultValue >
</parameter >

```

**Abbildung 16: Anlegen eines Freitextfeldes mit Vorbelegung**

Die Textvorbelegung für Freitextfelder kann in ProSAB mittels des Hilfsprogramms FreitextHelper (ein WYSIWYG-Editor) erstellt werden. Es erzeugt aus eingegebenen Texten und Tabellen aufbereiteten Code für das Freitextfeld in ProSAB. Der Code kann anschließend aus dem FreitextHelper in die Textlayout-Ansicht kopiert werden. Die Textvorbelegung wird jedoch innerhalb der WYSIWYG-Layout-Ansicht durch Oxygen schwer lesbar dargestellt.

Nachfolgend wird ein Freitextfeld (siehe Abbildung 17) mit vorbelegter Tabelle und Daten in der ProSAB-Bearbeitungsansicht dargestellt.



Risikoklassifizierung	aktuell	Vorjahr
Rating gesetzt (errechnet)	7 (5)	7 (5)
Punkte	5,22	5,02

Abbildung 17: Freitextfeld in ProSAB-Bearbeitungsansicht

Der Inhalt des Freitextfeldes erscheint in der zu erstellenden Entscheidungsvorlage im Abschnitt „V. Wirtschaftliche Verhältnisse“. Der Sachbearbeiter hat die Möglichkeit, anhand der Symbolleiste des Eingabefeldes, Formatierungen vorzunehmen sowie Tabellen einzufügen bzw. zu löschen. Weiterhin kann im Eingabefeld der Text beliebig bearbeitet werden.

Zu Beginn dieses Absatzes wurde bereits auf Abfragen hingewiesen, die im Template eine Steuerung der Elemente erlauben. Demnach werden im zu erstellenden Schriftstück nur bestimmte Absätze angezeigt. Die Notwendigkeit der unterschiedlichen Darstellung kann unter anderem aus der Antragsbearbeitung in ProSAB hervorgehen. Abfragen werden allerdings auch für Eingabefelder angewandt. So kann z.B. zum Freitextfeld aus Abb. 18 eine Bedingung erstellt werden, dass der Inhalt nur im Schriftstück erscheint, sofern das Feld gefüllt ist. Eine Vernachlässigung der Bedingung würde dazu führen, dass im Falle eines leeren Eingabefeldes im Schriftstück eine bzw. mehrere Leerzeilen erscheinen, wodurch das Schriftbild beeinträchtigt wird.

#### **4.5 Test des Templates „EEV für Darlehen“ in ProSAB**

Für die nähere Untersuchung des Templates, hinsichtlich der bereitgestellten Funktionen, sind in ProSAB das neue Template sowie die erstellten Bausteine zu administrieren. Anschließend ist ein neuer Förderantrag zu erstellen. In diesem Testantrag ist als Antragsteller eine Kreditnehmereinheit zu erfassen. Somit kann die korrekte Funktion des neuen Bausteins „Kreditnehmer“ überprüft werden. Weiterhin erfolgen Eingaben zur Finanzierung, den erwarteten Kosten und dem Objekt des Vorhabens. Nachdem die

notwendigen Daten im Antrag vorhanden sind, erfolgt die Erstellung der Entscheidungsvorlage im ProSAB-Schriftverkehr.

Es öffnet sich die ProSAB-Bearbeitungsansicht (siehe Abbildung 18). In dieser sind nun die Pflichtfelder und weitere Eingabefelder auszufüllen. Das erste Eingabefeld stellt eine Auswahlliste dar. Weiterhin erfolgen Eingaben in Textfelder bzw. es wird der vorgelegte Inhalt gelöscht, damit die korrekte Ausgabe der Zeilen und deren Abstände im endgültigen Schriftstück nachzuvollziehen ist.

The screenshot displays the ProSAB processing interface. At the top, there is a dropdown menu labeled 'bearbeitungstyp' with the option 'Bitte auswählen'. Below it, a list of categories is shown: 'organisierte Wohnungswirtschaft', 'private Vermieter', and 'Informationstechnologie'. A second dropdown menu is labeled 'Bitte Bearbeitungstyp auswählen!'. The form includes several text fields: 'Hausapparate: 4000', 'Engagementführer:', and 'Beschluss Dresden, den 14.08.2009'. A central text area contains the following text: 'Der Vorstand/Geschäftsbereichsleiter, Abteilungsleiter, Gruppenleiter beschließt', a numbered list '1.', '2.', and 'Es werden Beleihungswerte in Höhe von insgesamt € festgesetzt.'. Below this, there are fields for 'Datum, Unterschrift', 'letzte Beschluss:' (with a small 'letzter\_beschluss' label), 'Kompetenzträger:' (with a small 'kompetenztraeger' label), 'Gesamtengagement bei letztem Beschluss in TE:' (with a small 'gesamtengagement' label), and 'Charakter der Vorlage:' (with a small 'charakter' label). A 'Begründung für die vorgelegte Kompetenz:' field contains the text 'AO' (with a small 'begrueundung\_kompetenz' label). At the bottom, there is a 'Kreditart' dropdown menu set to 'Normalkredit' and a 'Bemerkung:' field with a small 'bemerkung' label.

**Abbildung 18: Erstellung der Entscheidungsvorlage in der ProSAB-Bearbeitungsansicht**

In Abbildung 19 ist ersichtlich, dass Festtext des Templates und Daten aus ProSAB lediglich nacheinander ausgegeben werden. Es besteht keine Formatierung und Ausrichtung des Textes. Die vorgenommenen Eingaben werden hinsichtlich ihrer Koordinaten erst im erzeugten PDF-Dokument berücksichtigt. Die Bearbeitungsansicht dient lediglich zur Steuerung der Dokumentabsätze bzw. der Vornahme von Eingaben. Nachdem die Felder der Ansicht ausgefüllt sind, kann das endgültige Schriftstück, die Entscheidungsvorlage, im PDF-Format erzeugt werden. Es öffnet sich ein neues Fenster, in dem das Resultat ersichtlich ist. Nun folgt eine genaue Überprüfung des Dokumentinhalts hinsichtlich Rechtschreibung, Formatierung, Ausrichtung und korrekte Funktionsweise der ProSAB-Keys und Bausteine. Angaben, die im Förderantrag getätigt wurden, müssen an den vorgesehenen Stellen in der Entscheidungsvorlage erscheinen. Zur detaillier-

---

ten Prüfung der Entscheidungsvorlage wird das Dokument ausgedruckt und mit der Vorgabe verglichen.

Sofern Fehler im Schriftstück festgestellt werden, müssen diese im Template behoben werden. Weiterhin sind in der ProSAB-Bearbeitungsansicht verschiedene Eingaben in den Auswahllisten vorzunehmen, damit die korrekte und gezielte Ausgabe der einzelnen Absätze, gemäß den Abfragen im Template, kontrolliert wird.

#### **4.6 Erfahrungen beim Einsatz WYSIWYG-fähiger XML-Editoren**

Die Untersuchung der ausgewählten XML-Editoren ergab, dass die Erstellung von ProSAB-Templates generell möglich ist. Nähere praktische Tests an den Produkten wiesen allerdings positive und negative Merkmale im Umgang mit der WYSIWYG-Layout-Ansicht auf.

Zunächst ist festzustellen, dass ein Stylesheet zur Verwendung der Ansicht durch den Nutzer bereitzustellen ist, sofern er nicht ein vorgegebenes XML-Format z.B. DocBook anwendet. Ein weiteres Manko, das durch ein eigenes CSS-Stylesheet in Editoren auftritt ist, dass die verfügbaren Funktionen in Symbolleisten und Menüs nur eingeschränkt bzw. gar nicht nutzbar sind. Für die vorgegebenen XML-Formate werden durch die Anwendungen zahlreiche Funktionen zur Formatierung bzw. zum Einfügen von Objekten, wie Tabellen und Grafiken bereitgestellt. Hierfür sind die Funktionen der Editoren an die gegebenen DTDs und XSDs der XML-Formate angepasst. Ein näherer Vergleich der DTDs mit der vorhandenen ProSAB-DTD zeigt Unterschiede in der Anzahl und Funktion der Elemente. XML-Formate wie DocBook und TEI dienen vor allem der Darstellung von Texten, z.B. in Büchern. Daher sind in der DTD zahlreiche Elemente zur Festlegung von Überschriften, Absätzen, Kapiteln enthalten. Dadurch ist der Text im XML-Dokument klar gegliedert und zu jedem Element können innerhalb der WYSIWYG-Layout-Ansicht bestimmte Formatierungen vorgenommen werden. Die ProSAB-DTD verwendet für die Festlegung von Text nur die Elemente *<staticText>* und *<textField>*. Hierdurch besteht bereits die erste Einschränkung. Es wird nicht zwischen Überschriften und normalen Absätzen im Schriftstück unterschieden. Allerdings verfügt die ProSAB-DTD über das Element *<textElement>*, welches der Formatierung von Texten dient. Ein Problem, das in ProSAB aus der einseitigen Nutzung des *<staticText>* und *<textField>* hervorgeht ist, dass Dokumentbestandteile (Überschrift,

Absatz) und ihre Formatierung durch das CSS-Stylesheet nicht unterschiedlich dargestellt werden können.

Zusätzlich verwendet das Reporting-Tool JasperReports zur Ausrichtung von Elementen, die z.B. Text und Bausteine enthalten, mehrere Koordinaten. Diese Attribute und Werte können nicht auf Elemente im CSS-Stylesheet angewandt werden. Hierdurch wird zugleich das Einfügen und Ausrichten von Grafiken und Tabellen in der WYSIWYG-Layout-Ansicht verhindert. Daher entspricht die momentane Darstellung von Templates in der WYSIWYG-Layout-Ansicht (siehe Abbildung 19) in gewisser Form der Textlayout-Ansicht.

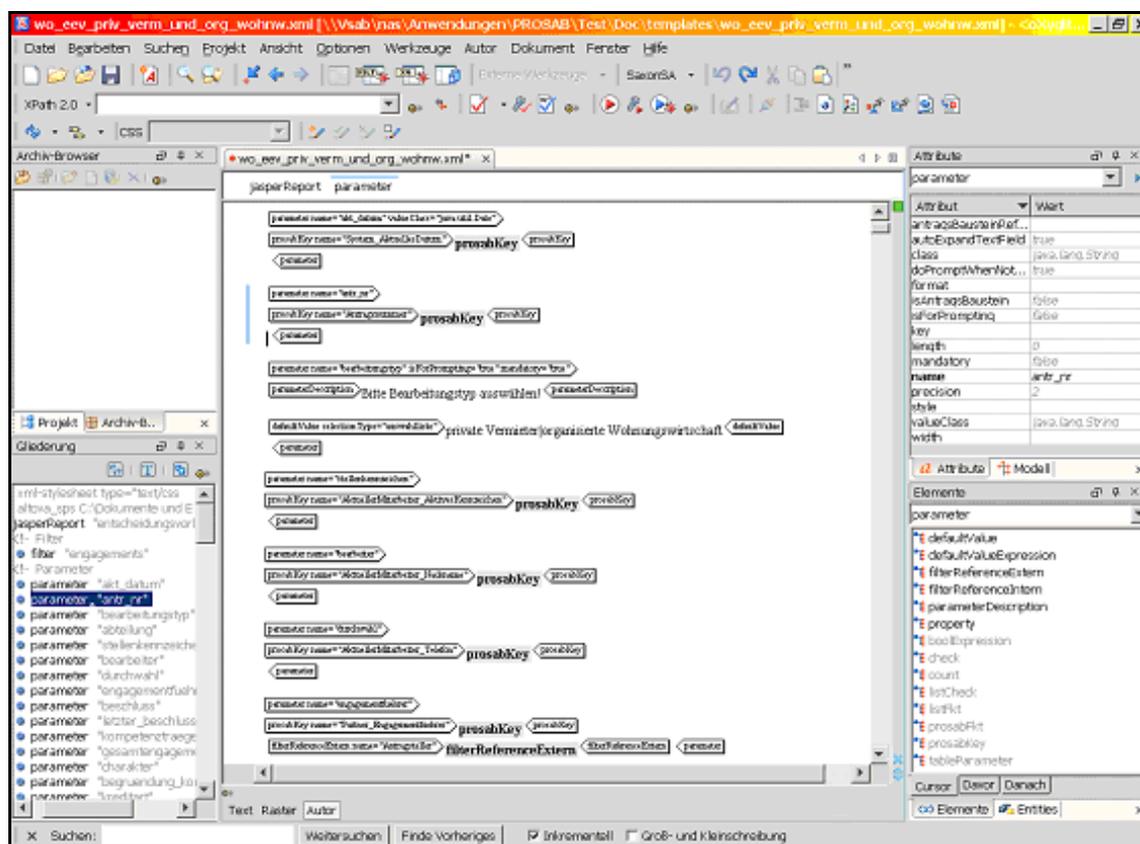


Abbildung 19: unvollständige WYSIWYG-Layout-Ansicht in Oxygen

Für die Nutzung WYSIWYG-fähiger Editoren spricht unter anderem die benutzerfreundliche Art, Elemente und Attribute einzufügen. Hierfür stehen z.B. in Oxygen Auswahllisten in der WYSIWYG-Layout-Ansicht und zusätzliche Eingabefenster am rechten Rand der Anwendung bereit. In diesem Zusammenhang ist positiv zu erwähnen, dass die Anwendungen importierte DTDs verwenden können. Daher werden nur Elemente und Attribute der DTD zum Einfügen bereitgestellt. Demzufolge sind Fehleingaben ausgeschlossen bzw. sie entstehen im geringeren Umfang. Dementsprechend ist das

---

Löschen von Elementen und Attributen ebenfalls gefahrlos durchzuführen. Alle Veränderungen am Template, die in der WYSIWYG-Layout-Ansicht erfolgen, werden gleichermaßen in der Textlayout-Ansicht umgesetzt. Fehler, die durch manuelles, unvollständiges Entfernen von Zeichenketten in der Textlayout-Ansicht entstehen können und die Wohlgeformtheit des Templates beeinträchtigen, sind in der WYSIWYG-Layout-Ansicht ausgeschlossen. Weiterhin können Elemente simpel per „Drag and Drop“ im Template verschoben und angeordnet werden.

Neben den Vorteilen beim Einfügen von Elementen zeichnen sich größere Probleme im Umgang mit Parametern und *<printWhenExpression>* Abfragen ab. Gegen die Verwendung der WYSIWYG-Layout-Ansicht spricht die Definition von Parametern, die zu Beginn des Templates erfolgt. Eine Aufzählung von Parametern würde den Sinn und Zweck des WYSIWYG, und zwar die Vorschau auf das endgültige Schriftstück erheblich beeinträchtigen. Gleichermäßen trägt die Aufführung von Abfragen, zwischen den Absätzen, zur Störung der Gesamtansicht des Dokuments bei. Diesbezüglich müssten die XML-Editoren über separate Fenster bzw. Ansichten verfügen, die eine Bearbeitung der Parameter und Abfragen ermöglichen. Am linken Rand des Editors Oxygen (siehe Abbildung 19) ist ein Fenster verfügbar, das alle verwendeten Elemente des Templates anzeigt. Es ermöglicht darüber hinaus das Einfügen und Löschen von Elementen. Weitere Eingaben zu den Parametern und Abfragen könnten alternativ in der Textlayout- oder Gridlayout-Ansicht erfolgen.

Im CSS-Stylesheet müssen die Elemente *<parameter>* und *<printWhenExpression>* für die Ausgabe innerhalb der WYSIWYG-Layout-Ansicht blockiert werden. Somit bleibt die Dokumentansicht unbeeinträchtigt von diesen Elementen.

## 5 Zusammenfassung und Ausblick

Nachfolgend werden die gewonnenen Ergebnisse der Untersuchung von WYSIWYG-fähigen Editoren aufgeführt und Möglichkeiten zu weiterführenden Betrachtungen aufgezeigt.

### 5.1 Erreichte Ergebnisse

Das Ziel der Arbeit war die Untersuchung der Einsatzfähigkeit von WYSIWYG-Layout-Ansichten in XML-Editoren. In diesem Zusammenhang wurden bekannte Anforderungen der SAB und der internen Applikation ProSAB als Bewertungskriterien formuliert, woraus ein Kriterienkatalog hervorging. Unter Berücksichtigung des Kriterienkataloges wurde eine Marktrecherche durchgeführt. Dabei wurden die ermittelten Editoren und deren Produktinformationen in Erhebungsbögen festgehalten. Es folgte die Bewertung der XML-Editoren anhand einer Nutzwertanalyse. Während der Durchführung wurde die Anzahl der zu untersuchenden Produkte anhand von K.O.-Kriterien oder einer zu geringen Bewertung reduziert. Für die weiterführenden Untersuchungen wurden die vier höchst bewerteten Produkte einer ausführlicheren Bewertung unterzogen. Bei den ausgewählten Produkten wurden die Benutzeroberfläche, Anwendungsfunktionen sowie die Softwarequalität untersucht. Die Analyse ergab, dass das Produkt Oxygen die größte Übereinstimmung mit den aufgestellten Anforderungen aufweist.

Daher wurde der Editor Oxygen für die prototypische Erstellung des ProSAB-Templates „Einheitliche Entscheidungsvorlage für Darlehen“ ausgewählt. Die Untersuchung des XML-Editors bezüglich der WYSIWYG-Layout-Ansicht erfolgte unter den vorhandenen Rahmenbedingungen wie ProSAB-Keys und JasperReports. Während der Erstellung des Templates war festzustellen, dass aufgrund der vorhandenen DTD einige Programmfunktionen des Editors nicht vollständig zur Verfügung standen. Zusätzlich waren innerhalb des Bearbeitungszeitraumes notwendige Vorleistungen für Teilbereiche des Templates noch nicht umgesetzt, da noch zusätzliche Abstimmungen mit dem Fachbereich ausstanden. Der geplante Prototyp des Templates konnte somit innerhalb des gegebenen Zeitraums nicht vollständig fertiggestellt werden.

Die durchgeführte Untersuchung sowie der praktische Einsatztest des Editors Oxygen in der WYSIWYG-Layout-Ansicht hat ergeben, dass eine bedeutsame Vereinfachung der Templateerstellung nicht möglich ist.

---

Die angebotenen Technologien und Arbeitsabläufe der Editoren korrespondieren nicht mit den Gegebenheiten in der SAB. Die Verwendung eines WYSIWYG-fähigen Editors zur Erstellung von ProSAB-Templates kann daher zum jetzigen Zeitpunkt nicht empfohlen werden. Für einen zukünftigen Einsatz sind Veränderungen an den angebotenen Editoren und den technologischen Vorbedingungen innerhalb der SAB erforderlich.

## **5.2 Ausblick**

Das geforderte Ziel einer vereinfachten Templateerstellung durch Anwender, deren Kenntnisse in XML begrenzt sind, wurde nicht erreicht. Für die Erstellung von ProSAB-Templates sind weiterhin fachliches Hintergrundwissen in XML sowie technologische Kenntnisse im ProSAB Umfeld erforderlich.

Die bestehende Lösung mit JasperReports zur Erstellung von Schriftstücken und deren Konvertierung in das PDF-Format stellt momentan für die SAB die beste Lösung dar.

Die untersuchten XML-Editoren stellen überwiegend Funktionen für DocBook und Text Encoding Initiative (TEI) bereit. Daher könnte eine Überprüfung durchgeführt werden ob solche Funktionen für die Zwecke der SAB nutzbar sind. Es ist aber zu beachten, dass eine Änderung der XML-Struktur den Austausch des XML-Parsers erfordert und ein hoher Migrationsaufwand (ca. 1700 ProSAB-Templates) besteht.

Für den Editor Oxygen wird durch den Hersteller die Möglichkeit eingeräumt, eigene Plugins in Java zu programmieren. Dazu müsste untersucht werden, ob eine Anpassung der WYSIWYG-Layout-Ansicht in Oxygen an die vorhandenen ProSAB-Bedingungen möglich ist.

Aufgrund dessen, dass die spezifischen ProSAB-Anforderungen vermutlich auch in Zukunft durch kein Softwareprodukt erfüllt werden, ist es vorstellbar, dass innerhalb der SAB ein eigener WYSIWYG-fähiger Editor entwickelt wird. Dieser könnte auf Grundlage der unentgeltlichen Editoren iReport oder JasperAssistant erstellt werden. Außerdem wäre eine genaue Anpassung des Editors an die Anforderungen von ProSAB und JasperReports möglich.

Da ProSAB innerhalb der SAB entwickelt und gepflegt wird, sollte eine weitergehende Beobachtung des Marktes und Bewertung der Technologien durchgeführt werden, um dem Ziel der vereinfachten Erstellung von Templates näher zu kommen.

## Glossar

JasperReports	Ein kostenloses Reporting-Tool zum Erzeugen von Dokumenten. Das Tool kann in Java-Applikationen eingebunden werden.
WYSIWYG	„What You See Is What You Get“, nach diesem Prinzip entspricht die Dokumentansicht am Bildschirm der Ausgabe des Dokuments durch den Drucker.
XML	Die Extensible Markup Language ist eine standardisierte Metasprache, die der Erstellung von Auszeichnungssprachen dient.
XML-Template, ProSAB-Template	XML-Dokumente, die als Schablonen für die Erstellung von Schriftstücken in Anwendungen fungieren.
ProSAB	Eine Java-Applikation der Sächsischen Aufbaubank -Förderbank-, die der internen Bearbeitung von Förderanträgen dient.
Stylesheets	Formatvorlagen für die Präsentation von Daten. Das Cascading Stylesheet (CSS) und Extensible Stylesheet Language Transformations (XSLT) sind Formatvorlagen, die z.B. der Darstellung von XML-Dokumenten in WYSIWYG-Editoren dienen.
Einheitliche Entscheidungsvorlage	Dokumentvorlage, die der Entscheidungsfindung über die Annahme bzw. Ablehnung eines Kreditantrags dient.

## Literaturverzeichnis

**BECKER, J./VERING, O./WINKELMANN, A. (2007):**

Softwareauswahl und -einführung in Industrie und Handel, Vorgehen bei und Erfahrungen mit ERP- und Warenwirtschaftssystemen; Berlin; Springer Verlag; 2007

**BOOS, K.-H./FISCHER R./SCHULTE-MATTLER H.(2004):**

Kreditwesengesetz-Kommentar zu KWG und Ausführungsvorschriften; 2. Aufl. München; Verlag C. H. Beck; 2004

**BOURRET, R. (2009):**

<rpbouret@rpbouret.com>; XML and Databases;  
URL: <<http://rpbouret.com/xml/XMLAndDatabases.htm#templatebased>>;  
verfügbar am 27.07.2009

**COMPUTERWOCHE (2009):**

<cw@computerwoche.de>; Computerwoche;  
URL: <<http://www.computerwoche.de/software/software-archiv/>>;  
verfügbar am 16.06.2009

**DANCIU, T. (2002):**

<teodord@hotmail.com>; The JasperReports Ultimate Guide Version 1.0;  
PDF-Dokument (2002)

**HAROLD, E. R./MEANS, W. S. (2005):**

XML in a Nutshell; 3 Aufl. Köln; O'Reilly Verlag; (2005)

**HEFFELFINGER, D. R. (2006):**

JasperReports for Java Developers, Create, Design, Format, and Export Reports with the World's Most Popular Java Reporting Library;  
Birmingham; Packt Publishing Ltd.; (2006)  
URL: <<http://www.infoap.utcluj.ro/bdate/jreports.pdf>>; verfügbar am 04.08.2009

**HERCZEG, M. (2009):**

Software-Ergonomie; Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme; 3. überarb. und erw. Aufl. München; Oldenbourg Wissenschaftsverlag GmbH; 2009

**IREPORT (2009):**

JasperForge; URL: <<http://jasperforge.org/projects/ireport>>; verfügbar am 23.07.2009

**JASPERASSISTANT (2009):**

<contact@jasperassistant.com>; JasperAssistant;  
URL: <<http://www.jasperassistant.com/>>; verfügbar am 24.07.2009

**JASPERREPORTS (2009):**

JasperReports; URL: <[jasperforge.org/jasperreports](http://jasperforge.org/jasperreports/)>; verfügbar am 17.06.2009

**JUSTSYSTEMS (2009):**

Justsystems;  
URL <[http://na.justsystems.com/content-xmetal-author](http://na.justsystems.com/content-xmetal-author/)>; verfügbar am 23.07.2009

**KLEIN, O. (2009):**

XML und Datenbanken;  
URL: <[http://www.urz.uniheidelberg.de/imperia/md/content/urz/programme/db\\_und\\_xml.pdf](http://www.urz.uniheidelberg.de/imperia/md/content/urz/programme/db_und_xml.pdf)>; verfügbar am 27.07.2009

**Kredithandbuch** der Sächsischen Aufbaubank -Förderbank- vom 09.07.2009

**LIQUID (2009):**

<sales@liquid-technologies.com>; Liquid XML Studio; URL: <<http://www.code-generator.com/>>; verfügbar am 22.07.2009

**OXYGEN (2009):**

<support@oxygenxml.com>; Oxygen  
URL: <[http://www.oxygenxml.com/wysiwyg\\_xml\\_editor.html](http://www.oxygenxml.com/wysiwyg_xml_editor.html)>;  
<<http://www.oxygenxml.com/doc/Oxygen-UserManual.pdf>>; verfügbar am 10.09.2009

**PARTL, H. (2009):**

<partl@mail.boku.ac.at>; XML - Extensible Markup Language;  
URL: <<http://www.boku.ac.at/html/einf/xmlkurz.html>>; verfügbar am 29.06.2009

**PARC (2009):**

<pr@parc.com>; Palo Alto Research Center;  
URL: <<http://www.parc.com/about/milestones.html>>; verfügbar am 16.07.2009

**PEPELS, W. (2006):**

Produktmanagement-Produktinnovation, Markenpolitik, Programmplanung, Prozessor-  
ganisation; 5. überarb. Aufl. München; Oldenbourg Wissenschaftsverlag GmbH; 2006

**SCHILLER B./TYTKO D. (2001):**

Risikomanagement im Kreditgeschäft-Grundlagen, neuere Entwicklungen und Anwendungsbeispiele; Stuttgart; Schäffer-Poeschel; 2001

**SCHÖNING, H. (2003):**

XML und Datenbanken-Konzepte und Systeme; München; Carl Hanser Verlag; 2003

**SERNA (2009):**

<info-general@syntext.com>; Syntext;

URL: <<http://www.syntext.com/products/serna/>>; verfügbar am 22.07.2009

**STOWASSER, S. (2006):**

Methodische Grundlagen der softwareergonomischen Evaluationsforschung; Aachen; Shaker Verlag; 2006

**STYLEVISION (2009):**

<eu-office@altova.com>; Altova StyleVision 2009 Benutzer-und Referenzhandbuch;

URL: <<http://www.altova.com/de/documents/StyleVisionEnt.pdf>>;

verfügbar am 22.07.2009

**SOFTGUIDE (2009):**

<uwe.annuss@softguide.de>; SoftGuide der Softwareführer;

URL: <<http://www.softguide.de/>>; verfügbar am 16.06.2009

**SOFTWARE (2009):**

Software-Marktplatz; URL: <[www.software-marktplatz.de](http://www.software-marktplatz.de)>; verfügbar am 16.06.2009

**SYDOCK (2009):**

Sydock; URL: <<http://www.sydock.com/asp/products/info/features.asp>>;

verfügbar am 23.07.2009

**TIEMEYER, E./FEIL, T. (2007):**

Handbuch IT-Management; 2. Aufl. München; Carl Hanser Verlag; 2007

**XMLMIND (2009):**

<xmleditor-info@xmlmind.com>; Pixware-XMLmind;

URL: <<http://www.xmlmind.com/xmleditor/>>; verfügbar am 22.07.2009

# Anlage 1

## Blatt 1

### Erhebungsbogen für WYSIWYG-fähige XML-Editoren

Nr.	Informationen					
1.1	Produktname		XMLmind	Serna Enterprise	XMLSpy 2009	Oxygen XML Editor
1.2	Version		4.4.0	4.1		10.2
1.3	Gründungsjahr des Anbieters		1993	1999	1992	1998
1.4	Firmensitz		Frankreich, Rambouillet	Seattle, Washington	Österreich, Wien	Rumänien, Craiova
1.5	Internetadresse des Produkts		http://www.xmlmind.com/	http://www.syntext.com/	http://www.altova.com/	http://www.oxygenxml.com/
1.6	Testinstallation		ja	ja	ja	ja
Nr.	Angaben zum Produkt und Anbieter	K.O. Kriterium				
2.1	Bekanntheitsgrad des Anbieters		Pixware	Syntext	Altova	Syncro Soft
2.2	Leistungsfähigkeit des Unternehmens - Anzahl Mitarbeiter		keine Angabe	keine Angabe	keine Angabe	- 20-40 Mitarbeiter
2.3	Anzahl der Lizenzen		keine Angabe	keine Angabe	- 3 Millionen Kunden (auf alle Produkte von Altova bezogen)	- einige Tausend
2.4	Releasehäufigkeit im Jahr		keine Angabe	keine Angabe	- ca. 2 mal pro Jahr	- 1 major Release/Jahr - 1 minor Release aller 3 Monate
2.5	Referenzen (vor allem Banken und Versicherungen)		- zahlreiche Unternehmen wie Fujitsu Siemens, General Motors	- zahlreiche Universitäten, Siemens, IBM	- 3 Millionen User - 100.000 Firmen - Deutsche Bank	- Zurich Insurances - Raiffeisen Kapitalanlage-Gesellschaft mbH
2.6	Benutzersupport, Benutzerhandbuch, Online-Hilfe, Wartung		- 125 €/Stunde Customization Service - auf Kundenwunsch Erstellung von customization packages - E-Mail-Support	- freier E-Mail-Support mit 48 Std. Antwortzeit - Telefon, Fax - 1 Jahr lang kostenlose Version-Updates für registrierte Kunden, nach dem Produktkauf - Support Package \$ 149 für 1 Jahr technischen-Support, inkl. Updates und major Releases - Savings per User \$ 50 Preisnachlaß und Mengenrabatt	- 30 Tage Support nach dem Erwerb eines Produktes - Online-Support mit 25% Aufpreis für 1 Jahr (20% ab dem 2 Jahr) erhältlich, inkl. aller minor und major Release - Kauf von Upgrades möglich - Online Handbuch	- E-Mail-, Telefon-, Fax-Support - Online Forum - Maintenance Pack für 1 Jahr, enthält alle minor und major Releases, kostet pro Lizenz € 73
2.7	Add-ons/Plugins		- Rechtschreibprüfung Deutsch - XSL-FO Prozessor - Konfigurationen - deutsche Menüführung	- Rechtschreibprüfung Deutsch	- IDE Integration - Prozessoren für XSLT und XSL-FO	- Programmierung von Plugins möglich
2.8	Preis/Lizenz		- Professional Edition € 230	- Enterprise Edition \$ 559	- Enterprise Edition € 499	- Professional Edition \$ 349

# Anlage 1

## Blatt 2

### Erhebungsbogen für WYSIWYG-fähige XML-Editoren

Nr.	Informationen						
1.1	Produktname		Liquid XML Studio - Graphical XML Editor	XMetaL Author Essential	Sydock document processor, Enterprise	iReport	JasperAssistant
1.2	Version		7.1.4	5.5	2.5.4.7	3.5.2	3.1.0
1.3	Gründungsjahr des Anbieters		2000	1979	keine Angabe	2004	keine Angabe
1.4	Firmensitz		United Kingdom, Bradford	United Kingdom, Baldock, Hertfordshire	keine Angabe	Kalifornien, San Francisco	Frankreich, Bourg-Les-Valence
1.5	Internetadresse des Produkts		<a href="http://www.code-generator.com">http://www.code-generator.com</a>	<a href="http://na.justsystems.com">http://na.justsystems.com</a>	<a href="http://www.sydock.com">http://www.sydock.com</a>	<a href="http://jasperforge.org">http://jasperforge.org</a>	<a href="http://www.jasperassistant.com">http://www.jasperassistant.com</a>
1.6	Testinstallation		ja	ja	ja	ja	ja
Nr.	Angaben zum Produkt und Anbieter	K.O. Kriterium					
2.1	Bekanntheitsgrad des Anbieters		Liquid Technologies	Just Systems	Sydock	Jaspersoft JasperForge	INFOLOGIC SA
2.2	Leistungsfähigkeit des Unternehmens - Anzahl Mitarbeiter		keine Angabe	- über 1000 Mitarbeiter - Jahresumsatz über \$ 110M	keine Angabe	- Entwicklung durch Community - 90.000 Mitglieder - über 350 Projekte	keine Angabe
2.3	Anzahl der Lizenzen		keine Angabe	- 2500 Kunden insgesamt (bezogen auf alle Produkte)	keine Angabe	- 8 Millionen Downloads der Business Intelligence Suite, diese enthält den iReport	keine Angabe
2.4	Releasehäufigkeit im Jahr		keine Angabe	keine Angabe	keine Angabe	keine Angabe	mehrere Releases pro Jahr
2.5	Referenzen (vor allem Banken und Versicherungen)		- zahlreiche Banken - Bank of America - Lehman Brothers	- Allianz - Deutsche Bank	keine Angabe	- 10.000 Handelskunden - Zurich Financial Services - Harland Financial Solutions	keine Angabe
2.6	Benutzersupport, Benutzerhandbuch, Online-Hilfe, Wartung		- umfangreiches Online- Handbuch - XML-Tutorial - 12 Monate Email-Support	- E-Mail-Support - Online-Forum - Online-Handbuch - Annual Maintenance Agreement ermöglicht 1 Jahr Online-, Telefon- Support, Updates und alle Release	- Online Forum - E-Mail-Support - 30 Tage freier Support - Support für 1 Jahr kostet 10% vom Kaufpreis	- Online Forum	- E-Mail-Support

# Anlage 1

## Blatt 3

### Erhebungsbogen für WYSIWYG-fähige XML-Editoren

Nr.	Informationen					
1.1	Produktname		XMLmind	Serna Enterprise	XMLSpy 2009	Oxygen XML Editor
Nr.	Allgemeine Systemeigenschaften	K.O. Kriterium				
3.1	Unterstützt Betriebssystem Windows 2000, XP, Vista	X	ja	ja	ja	ja
3.2	Applikation		ja	ja	ja	ja
3.3	Erstellung von XML-Dokumenten entspricht der W3C Spezifikation für XML		ja	ja	ja	ja
3.4	Unterstützt JasperReports		nein	nein	nein	nein
Nr.	Benutzeroberfläche, Bearbeitungsansichten	K.O. Kriterium				
4.1	Textlayout-Ansicht	X	ja	ja	ja	ja
4.2	Gridlayout-Ansicht		ja	keine Angabe	ja	ja
4.3	WYSIWYG-Layout-Ansicht	X	keine Angabe	keine Angabe	ja	ja
4.4	Menüsprache - deutsch (Priorität) - englisch		deutsch	deutsch	deutsch	deutsch
4.5	Seitenvorschau		keine Angabe	keine Angabe	ja	keine Angabe
Nr.	Entwurf, Dateneingabe und -bearbeitung	K.O. Kriterium				
5.1	Unterstützung von Schemas (vor allem DTD)	X	ja	ja	ja	ja
5.2	kontextsensitive Eingabemöglichkeiten		ja	ja	ja	ja
5.3	kontextsensitive Bearbeitungshilfen		keine Angabe	ja	ja	ja
5.4	Code-Snippets		keine Angabe	keine Angabe	ja	ja
5.5	Standardfunktionen		ja	ja	ja	ja
5.6	Unicode und andere Zeichensätze		keine Angabe	unterstützt Unicode	unterstützt Unicode	unterstützt Unicode
5.7	Einfügen von Grafiken		ja	ja	ja	ja
5.8	Versionierung von XML-Dokumenten (SVN)		nein	nein	nein	ja
5.9	Struktur und Umfang des generierten XML-Codes		keine Angabe	keine Angabe	keine Angabe	keine Angabe
Nr.	Prüfroutinen	K.O. Kriterium				
6.1	Validierung des XML-Templates im Editor	X	ja	ja	ja	ja
6.2	„On the fly“ Validierung		ja	ja	ja	ja
6.3	Validierung ein- und ausschalten		keine Angabe	ja	nein	ja
6.4	Prüfung auf Wohlgeformtheit	X	ja	ja	ja	ja
6.5	Syntaxanzeige im Code-view		keine Angabe	ja	nein	keine Angabe
6.6	Rechtschreibprüfung		ja	ja, mehrsprachig	ja, bis zu 18 Sprachen	keine Angabe

# Anlage 1

## Blatt 4

### Erhebungsbogen für WYSIWYG-fähige XML-Editoren

Nr.	Allgemeine Systemeigenschaften	K.O. Kriterium					
3.1	Unterstützt Betriebssystem Windows 2000, XP, Vista	X	ja	unterstützt XP und Vista	unterstützt Win 2000 und XP	unterstützt XP und Vista	ja
3.2	Applikation		ja	ja	ja	ja	ja, Eclipse Plugin
3.3	Erstellung von XML-Dokumenten entspricht der W3C Spezifikation für XML		ja	ja	ja	ja	ja
3.4	Unterstützt JasperReports		nein	nein	nein	ja	ja
Nr.	Benutzeroberfläche, Bearbeitungsansichten	K.O. Kriterium					
4.1	Textlayout-Ansicht	X	ja	ja	ja	ja	nein
4.2	Gridlayout-Ansicht		ja	ja	nein	nein	nein
4.3	WYSIWYG-Layout-Ansicht	X	nein	ja	nein	nein	nein
4.4	Menüsprache - deutsch (Priorität) - englisch		englisch	englisch	englisch	deutsch	englisch
4.5	Seitenvorschau		nein	ja	ja	ja	ja
Nr.	Entwurf, Dateneingabe und - bearbeitung	K.O. Kriterium					
5.1	Unterstützung von Schemas (vor allem DTD)	X	ja	ja	ja	ja	ja
5.2	kontextsensitive Eingabemöglichkeiten		keine Angabe	keine Angabe	keine Angabe	keine Angabe	nein
5.3	kontextsensitive Bearbeitungshilfen		ja	keine Angabe	keine Angabe	keine Angabe	nein
5.4	Code-Snippets		nein	keine Angabe	keine Angabe	keine Angabe	nein
5.5	Standardfunktionen		ja	ja	keine Angabe	ja	ja
5.6	Unicode und andere Zeichensätze		keine Angabe	ja	keine Angabe	ja	
5.7	Einfügen von Grafiken		keine Angabe	ja	ja	ja	ja
5.8	Versionierung von XML-Dokumenten (SVN)		nein	keine Angabe	keine Angabe	nein	nein
5.9	Struktur und Umfang des generierten XML-Codes		keine Angabe	keine Angabe	keine Angabe	keine Angabe	keine Angabe
Nr.	Prüfroutinen	K.O. Kriterium					
6.1	Validierung des XML-Templates im Editor	X	ja	ja	ja	ja	ja
6.2	„On the fly“-Validierung		keine Angabe	ja	keine Angabe	nein	
6.3	Validierung ein- und ausschalten		keine Angabe	keine Angabe	keine Angabe	keine Angabe	nein
6.4	Prüfung auf Wohlgeformtheit	X	ja	ja	ja	ja	ja
6.5	Hervorhebung von Syntaxfehlern		ja	keine Angabe	ja	ja	nein
6.6	Rechtschreibprüfung		keine Angabe	keine Angabe	ja	nein	nein

# Anlage2

## Blatt 1

### Bewertungsbogen für WYSIWYG-fähige XML-Editoren

Nr.	Informationen													
1.1	Produktname				XMLmind	Serna Enterprise	XMLSpy 2009	Oxygen XML Editor	Liquid XML Studio	XMetaL Author Essential	Sydock Enterprise	iReport	Jasper Assistant	
Nr.	Angaben zum Produkt und Anbieter	K.O. Kr.	GW Kr.	GW Gr.										
2.1	Bekanntheitsgrad Anbieter		15%	15%	10%	5%	15%	10%	8%	10%	5%	5%	5%	
2.5	Referenzen		20%		10%	10%	20%	15%	15%	15%	0%	15%	0%	
2.6	Benutzersupport		30%		25%	30%	25%	30%	20%	30%	25%	10%	10%	
2.7	Add-ons/Plugins		10%		10%	5%	10%	0%	0%	0%	0%	0%	0%	
2.8	Preis/Lizenz		25%		15%	10%	10%	10%	15%	5%	20%	25%	20%	
	<b>Summe</b>		<b>100%</b>		<b>70%</b>	<b>60%</b>	<b>80%</b>	<b>65%</b>	<b>58%</b>	<b>60%</b>	<b>50%</b>	<b>55%</b>	<b>35%</b>	
<b>Gesamtgewicht in Gruppe</b>					<b>10,50%</b>	<b>9,00%</b>	<b>12,00%</b>	<b>9,75%</b>	<b>8,70%</b>	<b>9,00%</b>	<b>7,50%</b>	<b>8,25%</b>	<b>5,25%</b>	
Nr.	Allgemeine Systemeigenschaften	K.O. Kriterium	Gewicht Kriterium	Gewicht Gruppe										
3.1	Unterstützt Betriebssystem Windows 2000, XP, Vista	X	70%	10%	70%	70%	70%	70%	70%	0%	0%	0%	70%	
3.4	Unterstützt JasperReports		30%		0%	0%	0%	0%	0%	0%	0%	0%	30%	30%
	<b>Summe</b>		<b>100%</b>		<b>70%</b>	<b>70%</b>	<b>70%</b>	<b>70%</b>	<b>70%</b>	<b>70%</b>	<b>0%</b>	<b>0%</b>	<b>30%</b>	<b>100%</b>
<b>Gesamtgewicht in Gruppe</b>					<b>7%</b>	<b>7%</b>	<b>7%</b>	<b>7%</b>	<b>7%</b>	<b>0%</b>	<b>0%</b>	<b>3%</b>	<b>10%</b>	
<b>Summe 1. und 2. Gruppe/Rangfolge</b>					<b>17,50% / 2</b>	<b>16,00% / 4</b>	<b>19,00% / 1</b>	<b>16,75% / 3</b>	<b>15,70% / 5</b>	<b>9,00% / 8</b>	<b>7,50% / 9</b>	<b>11,25% / 7</b>	<b>15,25% / 6</b>	
Nr.	Benutzeroberfläche	K.O. Kriterium												
4.1	Textlayout-Ansicht	X	30%	20%	0%	30%	30%	30%	/	/	/	/	/	
4.2	Gridlayout-Ansicht		15%		0%	0%	15%	15%	/	/	/	/	/	
4.3	WYSIWYG-Layout-Ansicht	X	30%		30%	30%	30%	30%	/	/	/	/	/	
4.4	Menüsprache - deutsch (Priorität) - englisch		15%		10%	7%	15%	15%	/	/	/	/	/	
4.5	Seitenvorschau		10%		0%	0%	10%	10%	/	/	/	/	/	
	<b>Summe</b>		<b>100%</b>		<b>40%</b>	<b>67%</b>	<b>100%</b>	<b>100%</b>	<b>/</b>	<b>/</b>	<b>/</b>	<b>/</b>	<b>/</b>	
<b>Gesamtgewicht in Gruppe</b>					<b>8,00%</b>	<b>13,40%</b>	<b>20,00%</b>	<b>20,00%</b>	<b>/</b>	<b>/</b>	<b>/</b>	<b>/</b>	<b>/</b>	

Anlage 2  
Blatt 2

Bewertungsbogen für WYSIWYG-fähige XML-Editoren

Nr.	Entwurf, Dateneingabe und -bearbeitung	K.O. Kriterium											
5.1	Unterstützung von Schemas (vor allem DTD)	X	20%	15%	/	20%	20%	20%	/	/	/	/	/
5.2	kontextsensitive Eingabemöglichkeiten		15%		/	7%	10%	15%	/	/	/	/	/
5.3	kontextsensitive Bearbeitungshilfen		15%		/	7%	10%	15%	/	/	/	/	/
5.5	Standardfunktionen und Short Cuts		10%		/	10%	10%	10%	/	/	/	/	/
5.7	Einfügen von Grafiken		15%		/	0%	0%	0%	/	/	/	/	/
5.9	Struktur und Umfang des generierten XML-Codes		15%		/	15%	15%	15%	/	/	/	/	/
<b>Summe</b>			<b>90%</b>		/	<b>59%</b>	<b>65%</b>	<b>75%</b>	/	/	/	/	/
<b>Gesamtgewicht in Gruppe</b>				/	<b>8,85%</b>	<b>9,75%</b>	<b>11,25%</b>	/	/	/	/	/	
Nr.	Prüfroutinen	K.O. Kriterium											
6.1	Validierung des XML-Templates	X	30%	20%	/	15%	30%	30%	/	/	/	/	/
6.2	„On the fly“ Validierung		15%		/	0%	0%	15%	/	/	/	/	/
6.4	Prüfung auf Wohlgeformtheit	X	25%		/	25%	25%	25%	/	/	/	/	/
6.5	Hervorhebung von Syntaxfehlern		20%		/	0%	0%	20%	/	/	/	/	/
6.6	Rechtschreibprüfung		10%		/	10%	10%	10%	/	/	/	/	/
<b>Summe</b>			<b>100%</b>		/	<b>50%</b>	<b>65%</b>	<b>100%</b>	/	/	/	/	/
<b>Gesamtgewicht in Gruppe</b>				/	<b>10,00%</b>	<b>13,00%</b>	<b>20,00%</b>	/	/	/	/	/	
Nr.	Softwarequalität	K.O. Kriterium											
7.1	Funktionalität		30%	10%	/	8%	15%	20%	/	/	/	/	/
7.2	Zuverlässigkeit		15%		/	8%	15%	15%	/	/	/	/	/
7.3	Benutzbarkeit		25%		/	15%	15%	15%	/	/	/	/	/
7.4	Effizienz		10%		/	10%	7%	7%	/	/	/	/	/
7.5	Änderbarkeit		10%		/	10%	10%	10%	/	/	/	/	/
7.6	Übertragbarkeit		10%		/	10%	15%	15%	/	/	/	/	/
<b>Summe</b>			<b>100%</b>	/	<b>61%</b>	<b>77%</b>	<b>82%</b>	/	/	/	/	/	
<b>Gesamtgewicht in Gruppe</b>				/	<b>6,10%</b>	<b>7,70%</b>	<b>8,20%</b>	/	/	/	/	/	

**Anlage 2**  
**Blatt 3**

**Bewertungsbogen für WYSIWYG-fähige XML-Editoren**

Nr.	Softwareergonomie	K.O. Kriterium											
8.1	Aufgabenangemessenheit		15%	10%	/	5%	10%	15%	/	/	/	/	/
8.2	Selbstbeschreibungsfähigkeit		20%		/	17%	15%	15%	/	/	/	/	/
8.3	Steuerbarkeit (Dialog)		20%		/	20%	20%	20%	/	/	/	/	/
8.4	Erwartungskonformität		15%		/	15%	15%	15%	/	/	/	/	/
8.5	Fehlertoleranz		10%		/	5%	7%	10%	/	/	/	/	/
8.6	Individualisierbarkeit		10%		/	5%	10%	10%	/	/	/	/	/
8.7	Lernförderlichkeit		10%		/	10%	10%	10%	/	/	/	/	/
<b>Summe</b>			<b>100%</b>		/	<b>77%</b>	<b>87%</b>	<b>95%</b>	/	/	/	/	/
<b>Gesamtgewicht in Gruppe</b>					/	<b>7,70%</b>	<b>8,70%</b>	<b>9,50%</b>	/	/	/	/	/
<b>Summe der Gruppen Gewichte</b>						<b>62,05%</b>	<b>78,15%</b>	<b>85,70%</b>					

# Anlage 3

Blatt 1

Vorgabe zur Erstellung der Entscheidungsvorlage

## Sächsische Aufbaubank - Förderbank -

### Entscheidungsvorlage

Abteilung:

Wohnungsbau

Bearbeiter:

M1/Muster

Hausapparat/e:

0123

Engagementführer:

### Beschluss

Dresden, den 26.10.2009

Der Kompetenzträger beschließt

3.

4.

Es werden Beleihungswerte in Höhe von insgesamt € festgesetzt.

Datum,

Unterschrift

letzter Beschluss: gemeint ist der letzte risikorelevante Beschluss

Kompetenzträger:

Gesamtengagement bei letztem Beschluss in T€

Charakter der Vorlage:

### Begründung für die vorgelegte Kompetenz:

Normalkredit

Neuausreichung Darlehen

Erhöhung / Bildung / Auflösung einer Einzelwertberichtigung

### 1. Kreditnehmer

Name:

Kunden-Nr.:

000-000000

Sitz:

Gründungsdatum:

Branche:

Konzern-Nr.:

0000000

### Kreditnehmereinheit gemäß § 19 Abs. 2 KWG

nein

ja Bezeichnung:

# Anlage 3

## Blatt 2

### Vorgabe zur Erstellung der Entscheidungsvorlage

#### I. Engagementübersicht des Kreditnehmers/ der Kreditnehmereinheit

per **Datum**

Ge- schäftsart	Bestand				Antrag	Konditionen			Neu
	Risiko- betrag	davon Auszah- lungsrest	davon Zins- rück- stand	davon Tilgung- rück- stand	Verände- rung (+/-) (Risiko- betrag)	Zins/ Mar- ge in %	Art/ Tilg. in %	ZB Jahre	Risiko- betrag neu
	(in €)	(in €)	(in €)	(in €)	(in €)				(in €)
Darlehen Bestand	0	0	0	0	0		---	---	0
Neuzusage KfW	0	0	0	0	0				0
Neuzusage FED	0	0	0	0	0				0
<b>Gesamt- obligo</b>	0	0	0	0	0		---	---	0
ausgebuchte Kredite	0	---	---	---	0		---	---	0
Obligo Freistaat Sachsen	0	0	---	---	0		---	---	0
nachrichtlich Zuschüsse	0	0	---	---	0		---	---	0
<b>EWB</b>	0	---	---	---	0		---	---	0

\* Bei Bestandsdarlehen gilt die derzeitige kap.-gew. Ø Marge \*\*A-Annuitätendarlehen; R-Ratendarlehen; F-Festdarlehen

Genehmigungslage bzgl. der Rückstände:

Stundung / Kündigung

Rückstände dauerhaft seit: Monat / Jahr

z.B. geduldete Überziehung /

Gläubigerstruktur per **Datum** (nur anzugeben bei Engagements der Gruppen WO10/20):

in %	vor Entscheidung	nach Entscheidung
Bank A		
Bank B		

Evidenzmeldung für: **Name des Kreditnehmers**

Stichtag:	Gesamtver- schuldung in T€	Verschuldung SAB in T€	Anteil SAB in %	Anzahl Kredit- geber

#### II. Rechtliche Verhältnisse

(bitte nachstehendes Modul ggf. duplizieren)

Name:	Zeile kann entfallen, sofern nur ein Unternehmen als KN
Rechtsform:	
Stammkapital / Grundkapital / Genossenschaftsguthaben in T€	
per:	
Unternehmensgegenstand:	entfällt bei Wohnungsunternehmen

# Anlage 3

## Blatt 3

### Vorgabe zur Erstellung der Entscheidungsvorlage

Geschäftsführer/Vorstand:	
Gesellschafter/Beteiligungshöhe	

### III. Sicherheiten

lfd. Nr.	Sicherheitenart	Nominalwert in T€	Sicherungswert in T€	nachrangig besichert in T€
<b>Bestand</b>				
...	Grundsschulden	0.00	0.00	0.00
...	Bürgschaft FS Sachsen	0.00	0.00	0.00
<b>Σ Bestand</b>		0.00	0.00	0.00
<b>Neuantrag/ zu entscheidendes Engagement</b>				
...		0.00	0.00	0.00
...		0.00	0.00	0.00
<b>Σ Neuantrag</b>		0.00	0.00	0.00
<b>Gesamt SAE</b>		0.00	0.00	0.00

Die letzte Sicherheitenplausibilisierung erfolgte am **TT.MM.JJJJ**.

Für die Grundschuld(en) wird ein weiter Sicherungszweck vereinbart.  
*alternativ:*  
 Grundschuld(en) in Höhe von € ... werden mit engem Sicherungszweck vereinbart.

Folgende(r) Beleihungswert(e) wird/werden festgesetzt:

Grundstücksbezeichnung	Beleihungswert		Ø nachhaltig erzielbare Mieteinnahmen (Nettokaltmiete)	x-faches der nachhaltig erzielbaren Mieteinnahmen
	in €	in €/m <sup>2</sup>		
Ort/ Str./ Hausnr./ Flst.				

In den nachhaltig erzielbaren Mieteinnahmen sind Stellplatzmieten i.H.v. T€ p. a. enthalten.

# Anlage 3

## Blatt 4

### Vorgabe zur Erstellung der Entscheidungsvorlage

#### IV. Verwendungszweck/Vorhabensdarstellung

Maßnahme/Vorhaben:	
Bau-/Investitionsort:	
Objektdaten:	Anzahl WE / Anzahl m <sup>2</sup>
Ø Sollnettokaltmiete:	0.00 €/ m <sup>2</sup>
Ø Leerstand:	%

alternativ bei Engagements der Gruppen WO 30 bis WO 50:

Grundstücksbezeichnung	Anzahl		Anzahl m <sup>2</sup>		Ø Sollnettokaltmiete in €/m <sup>2</sup>	Leerstand in %	Bauzustand
	WE	GE	Wfl.	Nfl.			
Ort/ Str./ Hausnr./ Flst.							saniert / teilsaniert / unsaniert

Gesamtkosten/Gesamtinvestition		Gesamtfinanzierung	
Bezeichnung	€	Bezeichnung	€
x	0.00	x	0.00
x	0.00	x	0.00
x	0.00	x	0.00
<b>Summe</b>	<b>0.00</b>	<b>Summe</b>	<b>0.00</b>

(Tabelle löschen sofern dem Beschluss ein abweichender Verwendungszweck zu Grunde liegt; Drittbankablösung etc.)

durchschnittliche Gesamtkosten in €/m<sup>2</sup>: 0.00

bei WO 10 /20: kurze Einschätzung des Standortes ergänzen  
Begründung für Ansatz der Mindestmarge sofern relevant.

# Anlage 3

## Blatt 5

### Vorgabe zur Erstellung der Entscheidungsvorlage

#### Objektwirtschaftlichkeitsberechnung

		Investitions- objekte s.o.	
Ist-Nettokaltmieten p. a. (Ø % Leerstand)	T€	0.00	
./ BWK p. a. (13 €/m²)	T€	0.00	
./ Abschlag für Mietausfallwagnis (10 %)	T€	0.00	
= aus dem Objekt tragbarer Kapitaldienst p. a.	T€	0.00	
./ Kapitaldienst Vorranggläubiger p. a. (8 % auf GS-Nominalbetrag T€)	T€	0.00	
= Kapitaldienstgrenze für SAB-Forderungen p.a.	T€	0.00	
./ Kapitaldienst SAB p. a. (8 % auf Risikobetrag)	T€	0.00	
= Überschuss / Fehlbetrag p. a.	T€	0.00	

alternativ bei der Ermittlung der EWB (private Vermieter):

#### Begründung des erwarteten Erlöses:

		Sicherungsobjekte	
Ist-Nettokaltmieten p. a. ( % Leerstand)	T€	0.00	
./ BWK p. a. (13 €/m²)	T€	0.00	
./ Abschlag für Mietausfallwagnis (10 %)	T€	0.00	
= aus dem Objekt tragbarer Kapitaldienst p. a.	T€	0.00	
kapitalisiert mit 7% p. a.:			
<b>= nachhaltig finanzierbarer Betrag</b>	<b>T€</b>	<b>0.00</b>	
./ Kosten für Reparaturrückstau / Instandsetzung	T€	0.00	
<b>= erwarteter Erlös (facher JRE)</b>	<b>T€</b>	<b>0.00</b>	

#### Ermittlung der Höhe des EWB-Bedarfs:

##### Berechnungsgrundlage für aktuelle Sicherheitenbewertung: erwarteter Erlös

Inanspruchnahme	T€	0.00	
erwarteter Erlös	T€	0.00	
./ Vorlasten	T€	0.00	
./ Gerichtskostenpauschale	T€	0.00	
./ Maklercourtage 3,57%	T€	0.00	
+ sonstige werthaltige Sicherheiten	T€	0.00	
<b>= werthaltige Sicherheiten</b>	<b>T€</b>	<b>0.00</b>	% der Inanspruchnahme
<b>nicht werthaltig gesichert</b>	<b>T€</b>	<b>0.00</b>	% der Inanspruchnahme
bereits gebildete EWB	T€	0.00	
<b>EWB-Bedarf</b>	<b>T€</b>	<b>0.00</b>	

Eine Rotmeldung in Höhe von T€ ... ist erfolgt in Monat/Jahr.

# Anlage 3

## Blatt 6

### Vorgabe zur Erstellung der Entscheidungsvorlage

#### V. Wirtschaftliche Verhältnisse

Offenlegung ist erforderlich wegen:  §18 KWG  freiwillig  
 Personalkredit > 250T€  nicht erforderlich

Engagement ist offen gelegt:  ja  nein bis  
 TT.MM.JJJJ

vorliegende Unterlagen:  Aufzählung bitte auf die wesentlichen Unterlagen begrenzen

Risikoklassifizierung	aktuell	Vorjahr	Fachrating Wohnen	aktuell	Vorjahr
Rating gesetzt (errechnet)	7 (5)	7 (5)	Gemeinde	4,7	4,9
Punkte	5,22	5,02	struktur. Leerstand	10,7%	11,8%

#### Darstellung des Wohnungsbestandes: *(nur anzugeben bei Engagements der Gruppen WO10/20)*

Stichtag:	Jahr	Vorjahr
Bestand	x WE / y m <sup>2</sup>	x WE / y m <sup>2</sup>
Leerstand in %	0.00 % (x WE)	0.00 % (x WE)
Sanierungsstand	0.00 % vollsaniert	0.00 % vollsaniert
	0.00 % teilsaniert	0.00 % teilsaniert
	0.00 % unsaniert	0.00 % unsaniert

bei Gruppen WO10/20: Sachstand / Planung zum Rückbau ergänzen

#### Zusammenfassung

- bei Privatpersonen: Einkommens- und Vermögenslage, insbesondere freie (unbelastete) Vermögenswerte, Kapitaldienstfähigkeit
- bei Unternehmen: Hauptumsatzbereich, Ertragslage, Finanz-/Vermögenslage, Liquiditätslage, Kapitaldienstfähigkeit des Unternehmens
- ggf. Objektwirtschaftlichkeit der SAB-Objekte
- bitte keine ausführliche Bonitätsanalyse, sondern Zusammenfassung der Ergebnisse der Bonitätsanalyse (Wertung!)
- Aussage zur Risikovorsorge treffen

# Anlage 3

## Blatt 7

### Vorgabe zur Erstellung der Entscheidungsvorlage

#### Bewertung der nachhaltigen Kapitaldienstfähigkeit

Kennzahl	gemäß Bilanz-analyse		gemäß Planung		
	Jahr	Bran- che	Jahr	Jahr	Jahr
Grenzkapitaldienstfähig- keit in %	0.00	0.00	0.00	0.00	0.00
cash flow in T€	0.00	---	0.00	0.00	0.00
cash flow in €/m <sup>2</sup> Nfl. p.m.	0.00	0.00	0.00	0.00	0.00
Tilgung in T€	0.00	---	0.00	0.00	0.00
Tilgung in %	0.00	0.00	0.00	0.00	0.00

- kritische Würdigung der Planzahlen, Bewertung der nachhaltigen Kapital dienstfähigkeit

#### VI. Besonderheiten/Auflagen

- Hier sind u.a. Auflagen aufzuführen, die nicht standardgemäß gelten.

- Bei Engagement mit einer Bürgschaft 12.2 des Freistaates Sachsen ist hier die weitere Verfahrensweise zu regeln (z.B. zur Kenntnis SMF; Zustimmung SMF...)

#### VII. Votum/Zusammenfassung/Engagementstrategie

Chancen	Risiken

Die Entscheidungsvorlage steht im Einklang mit der Risikostrategie der SAB.

Votum:

Engagementstrategie:

GBL			EF*			AL		GL		SB1		SB
-----	--	--	-----	--	--	----	--	----	--	-----	--	----

\* Engagementführer

## Anlage 3

### Blatt 8

## Vorgabe zur Erstellung der Entscheidungsvorlage

### Legende:

1.  Festtext: Text ist definierter Bestandteil des ProSAB-Template
2.  Baustein: Integration bestehender bzw. Neuanlage von Bausteinen, Daten werden zum Teil aus der Datenbank entnommen bzw. es erfolgen Benutzereingaben
3.  ProSAB: Die benötigten Daten werden mittels Listen und Keys durch ProSAB bereitgestellt
4.  Textfeld: Kann optional bzw. zwingend eine Texteingabe vom Sachbearbeiter erfordern
5.  Auswahlliste: Ein Eingabefeld, das optional bzw. zwingend vom Sachbearbeiter eine Eingabe fordert

## **Danksagung**

Mein ganz besonderer Dank gilt Herrn Prof. Dr. Wilfried Schubert an der Hochschule Mittweida (FH) für die Betreuung der Bachelorarbeit und die Unterstützung während der Anfertigung.

Des Weiteren danke ich Herrn Andreas Haake, stellv. Abteilungsdirektor der Abteilung Informationstechnologie bei der Sächsischen Aufbaubank-Förderbank als Zweitgutachter dieser Arbeit.

# **Selbstständigkeitserklärung**

Ich erkläre, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Dresden, 25.09.2009