

Web-basierte Beschaffung

Sebastian Kaiser, Eric Kuhnt, Stefanie Lemcke, Ulrike Lucke

Lehrstuhl Komplexe Multimediale Anwendungsarchitekturen
Institut für Informatik
Universität Potsdam
August-Bebel-Str. 89
14482 Potsdam
vorname.nachname@uni-potsdam.de

Abstract: Beschaffungsvorgänge sind aufwändig und komplex, insbesondere papiergebundene Varianten. Unterstützende Software ist jedoch teuer. Daher wurde an der Universität Potsdam eine Online-Beschaffungslösung implementiert, die in diesem Beitrag vorgestellt wird.

1 Motivation und Einleitung

Ein zentraler Bereich in der Hochschulverwaltung ist das Beschaffungswesen [ZKB12], das durch komplexe und zeitintensive Prozesse gekennzeichnet ist. Traditionell werden Beschaffungsvorgänge in Papierform abgewickelt, d.h. Formulare werden manuell befüllt, unterschrieben, abgestempelt, auf dem Postweg versandt, auf der Gegenseite wieder abgetippt, etc. Diese Vorgehensweise ist langwierig und fehleranfällig. Umfangreiche, teilweise unübersichtliche Formulare und vielfältige Verwaltungsvorschriften erschweren eine zügige Bearbeitung. So werden häufig Kostenstellen unvollständig oder fehlerhaft angegeben, Wertgrenzen nicht beachtet oder bestehende Rahmenverträge umgangen. Im schlimmsten Fall können dadurch Rückforderungen seitens der Aufsichtsbehörden bzw. klagender Konkurrenten entstehen.

Durch eine digitale Datenhaltung und netzbasierte Kommunikation können einige dieser Verfahren unterstützt und vereinfacht werden. Auch anschließende Prozesse wie die Zahlbarmachung von Rechnungen oder die Inventarisierung beschaffter Güter können so gut unterstützt werden. Ein zusätzlicher Mehrwert ist möglich, indem z.B. die Kataloge ausgewählter Anbieter direkt in das Beschaffungssystem integriert werden.

Dieser Bedarf wurde auch an der Universität Potsdam gesehen. In Zusammenarbeit mit dem Dezernat für Haushalt und Beschaffung und dem Rechenzentrum der Hochschule wurde ein Web-basiertes System für einen digitalen Beschaffungsprozess entwickelt. Der vorliegende Artikel beschreibt die Herangehensweise und die Ergebnisse dieses Projektes. Die Erkenntnisse sind (ggf. unter Beachtung kleinerer, lokaler Besonderheiten) auf andere Hochschulen übertragbar.

Das Vorhaben wurde gemäß dem Konzept des Service Learning umgesetzt [AM10]. Zunächst wurde im WS 2011/12 durch eine Gruppe Studierender die derzeitige Situation erfasst und analysiert. Der gesamte Prozessablauf wurde modelliert, die Anforderungen an das System wurden bei verschiedenen Beteiligten erhoben und ein rudimentärer Demonstrator des Laufzeitverhaltens dieses Prozessmodells entstand. Anschließend wurde im SS 2012 von einer erweiterten Gruppe Studierender die Implementierung vorgenommen. Dafür wurden bestehende Lösungen zur Prozessunterstützung, für das Formularmanagement und zur Systemintegration analysiert. Daraus wurden ein auf die Bedürfnisse und Anforderungen der Hochschule ausgerichtetes Datenmodell und eine Systemarchitektur entworfen. Die ausgewählten Frameworks wurden zusammengeführt und weiterentwickelt. Der aufgenommene Prozess wurde an den digitalen Ablauf angepasst und implementiert.

Der Beitrag geht zunächst auf die bestehenden Rahmenbedingungen aus technischer und organisatorischer Perspektive sowie auf bestehende Lösungen in diesem Umfeld ein. Auf dieser Basis werden die Systemkonzeption und die entstandene Beschaffungslösung beschrieben. Abschließend werden der erreichte Stand und die noch ausstehenden Arbeiten zur Produktivsetzung des Systems dargelegt.

2 Rahmenbedingungen

2.1 Organisatorischer Kontext

Beschaffungsvorgänge sind in der Regel komplexe Prozesse, welche durch viele Vorschriften und Formulare geprägt und durch Aktionen vieler beteiligter Personen beeinflusst werden.

Der Beschaffungsprozess der Universität Potsdam ist in die beiden Bereiche *Besteller* sowie *Haushalt und Beschaffung* gegliedert. Die Besteller sind im Allgemeinen Mitarbeiter, die an Fakultäten und Einrichtungen der Hochschule angestellt sind. Für eine Bestellung oder eine Auftragsvergabe muss ein Beschaffungsantrag ausgefüllt werden, welcher zudem durch einen Kostenstelleninhaber, eine Person mit Verwaltungsbefugnis über ein ihr unterstelltes Kontingent, genehmigt werden muss. Zusätzlich wird der Antrag durch einen Verwaltungsleiter oder ähnlich gestellten Mitarbeiter dezentral geprüft und genehmigt.

Im Dezernat Haushalt und Beschaffung (D4) wird dann dieser Antrag auf Einhaltung der Vorschriften geprüft, ausgeschrieben und vergeben. Somit fungiert das Dezernat im Namen der Hochschule als Auftraggeber. Zudem wird hier der ganze Vorgang dokumentiert und archiviert.

Für die Auftragsvergabe sind also mindestens vier Personen beteiligt; für den Bezahlvorgang sind es ebenso viele. Die Kommunikation ist dabei hauptsächlich über Formulare geregelt und muss koordiniert werden.

2.2 Technologischer Kontext

Der konventionelle, papiergebundene Beschaffungsprozess wird durch die Kommunikation per Hauspost vor allem durch lange Wartezeiten und intransparente Informationspolitik beherrscht. Durch eine manuelle Übernahme der Daten in die jeweils benutzten Systeme entstehen nicht nur eine redundante Datenhaltung und eine ineffiziente Zeitnutzung, sondern es werden zudem Fehler bei der Datenübertragung unterstützt.

Dies kann jedoch durch ein hochschulweites rechnergestütztes System und eine zentrale Datenbank verhindert werden. Die bereits gespeicherten Daten können nicht nur prozessintern weiter verarbeitet werden, indem beispielsweise Formulare mit vorhandenen Informationen vorausgefüllt werden, sondern ermöglichen auch eine prozessübergreifende Unterstützung in Form von Statistiken.

Zudem ist eine IT-gestützte Ablaufplanung durch eine Workflow-Engine möglich, welche die Koordination zwischen den beteiligten Personen übernimmt und anhand der getroffenen Entscheidungen den nächsten Schritt einleitet. Durch eine Umsetzung des Prozesses in einer Modellierungssprache kann diese hoch dynamische Komponente schnell und ohne weiterführende Änderungen im System verändert werden.

Eine gut geeignete und daher weit verbreitete Umsetzung ist die Modellierung mit BPMN (Business Process Model and Notation). Es wird neben der technischen Ausführung in einer Workflow-Engine eine grafische Modellierungsunterstützung geboten, sodass auch nicht-technisch bewanderte Mitarbeiter in der Lage sind, den Prozess zu verstehen und Anpassungen vorzunehmen [He12].

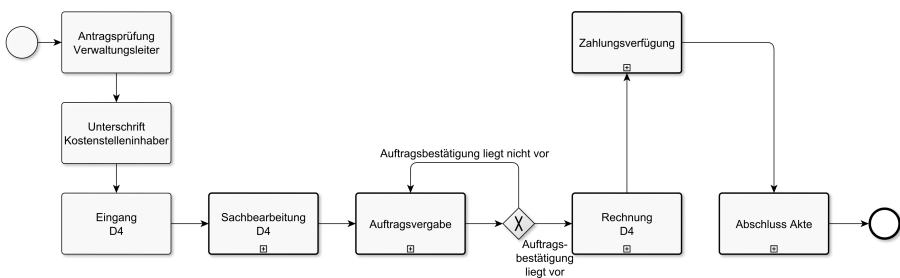


Figure 1: Vereinfachte Darstellung der Grundschrirte im Beschaffungsprozess

Die BPMN-Definition des Beschaffungsvorganges ist dabei in mehrere Prozessdiagramme gegliedert. Der Oberprozess (Abbildung 1) gibt die grobe Struktur vor. Logisch zusammengehörende Prozessschritte, vor allem im Bereich Haushalt und Beschaffung, wurden in Unterprozessen zusammengefasst, um eine bessere Übersicht zu gewährleisten. In diesem Beitrag kann nicht der gesamte Prozess in allen Details dargestellt werden; er besteht aus insgesamt 27 einzelnen Aktivitäten sowie diversen Akteuren und Verzweigungen. Da es sich im Falle des Beschaffungsprozesses um einen durch menschliche Aktivitäten gesteuerten Vorgang handelt, wurden hauptsächlich sogenannte User-Tasks verwendet, in welchen die Information des ausführenden Bearbeiters beziehungsweise der Gruppe, sowie die zu bearbeitenden Formulare oder auszuführenden Tätigkeiten hinterlegt

sind. Aufgaben, welche das System auszuführen hat, beispielsweise das Setzen von Variablen mit Werten aus der Datenbank, werden über sogenannte Service-Tasks realisiert.

2.3 Bestehende Ansätze

Im Zuge des Projektes wurden auch Recherchen zu bestehenden Systemen im Bereich elektronische Beschaffung durchgeführt. Dabei wurden mehrere Systeme näher betrachtet. Eine Lösung im Bereich E-Government und E-Procurement wird durch die cosinex GmbH bereitgestellt. Das System besteht aus drei Kernmodulen [co13]: dem Vergabekatalog, dem Vergabemarktplatz und dem Vergabemanagementsystem. Der Vergabekatalog unterstützt den Aufbau eines internen Bestell- und Katalogsystems. Neben der Darstellung von Produktkatalogen verschiedener Hersteller können auch mehrere Rahmenverträge abgebildet werden. Der zweite Kernbereich ist der Vergabemarktplatz, welcher öffentliche Vergabeverfahren von der Bekanntmachung über Bereitstellung von Verdingungsunterlagen bis zur Bieterkommunikation und der elektronischen Vergabe unterstützen soll. Es wird die Realisierung von Vergabemarktplätzen unterstützt und die Trennung des Systems in Vergabezentralen und dezentralen Vergabemarktplätzen erlaubt. Als letztes Kernmodul wird das Vergabemanagementsystem angeboten, welches den Vergabeprozess strukturieren und von der Bedarfsermittlung bis hin zur Zuschlagserteilung unterstützen soll. Zudem wird eine elektronische Vergabeakte angelegt, welche einen transparenten Vergabeprozess fördert und Verantwortlichen erlaubt, einen Überblick über die Vergabeverfahren zu erhalten.

Weitere Anbieter im Bereich elektronische Beschaffung sind auszugsweise Rödl & Partner [Rö13] und GISA GmbH [GI13].

Aufgrund finanzieller Zwänge wurde sich jedoch für eine Eigenentwicklung im Rahmen eines studentischen Projektes entschieden. Der damit entstehende zeitliche Verzug wurde angesichts des noch funktionierenden Altsystems toleriert. Zwar stehen den Anschaffungskosten und eventuellen Lizenzkosten für eine kommerzielle Lösung auch Kosten für die Administration und Wartung der Eigenentwicklung entgegen. Es ist jedoch anzunehmen, dass diese Kosten geringer sein werden. Des Weiteren stehen an Universitäten oftmals mehr Gelder für Personal als für Software zur Verfügung. Einem möglicherweise erhöhtem Wartungsaufwand wurde durch enge Einbeziehung des Rechenzentrums entgegengewirkt.

Ein weiterer Punkt, der für eine Eigenentwicklung sprach, ist die Individualität und Komplexität des Prozesses, welcher durch den Umgang mit sehr vielen und unterschiedlichen Formularen geprägt ist. Das System ist somit speziell auf die Abläufe und Probleme des Vorgangs zugeschnitten. Durch eine vollständige Administration im eigenen Haus können mögliche Änderungen im Prozessablauf flexibel durch den Systemadministrator vorgenommen, sowie auf entstehende Probleme schnell reagiert werden ohne auf Support von externen Dienstleistern warten zu müssen.

3 Konzeption und Umsetzung

Ausgehend von dem im vorigen Abschnitt beschriebenen Prozessmodell erfolgte die Umsetzung des Online-Beschaffungssystems, die hier getrennt nach Backend und Frontend erläutert wird.

3.1 Backend

Technologien

Die Universität Potsdam weist wie die meisten Hochschulen eine gewachsene heterogene IT-Landschaft auf, in welcher eine Vielzahl komplexer Prozesse durch unterschiedlichste Systeme abgebildet werden. Hierbei stellt die transparente und leistungsstarke Integration spezialisierter Systeme einen kritischen Faktor bezüglich der Wartbarkeit und Leistungsfähigkeit der IT-Landschaft dar. Diese Umfeldbedingungen treffen auch für das Beschaffungswesen zu.

Aus diesem Grund steht die Entwicklung eines leichtgewichtigen und anpassungsfähigen Systems auf Basis von quelloffenen Lösungen im Vordergrund der Konzeption und Umsetzung. Auf Basis der beiden Technologien Apache Karaf¹ und Apache Camel² wurde eine einfache, aber flexible Integrationsplattform [IAZ11] konzipiert, welche verschiedene Standards zur Bewältigung von Integrationsproblemen vereint. Hierbei wurde bewusst auf die Implementierung eines vollständigen Enterprise Service Bus zu Gunsten eines transparenten Systems verzichtet.

Apache Karaf ist eine leichtgewichtige OSGi-Plattform [CW13], die sich als Schlüsseltechnologie vieler weiterer Software-Lösungen bewährt hat. Karaf integriert mit Apache Aries eine solide Enterprise Implementierung, sodass die Entwicklung von lose gekoppelten Komponenten standardisiert möglich wird.

Bei Apache Camel handelt es sich um ein quelloffenes Integrationsframework basierend auf dem Konzept der Message Exchange Patterns. Ein sogenannter Routing-Engine-Buildermöglicht die Definition von Routen durch das System mittels einer domänen-spezifischen Sprache. Routen sind Regeln, die festlegen, von welchen Quellen Nachrichten akzeptiert, wie diese verarbeitet und wohin diese versendet werden sollen. Auf diese Weise wird eine higher-level Abstraktion bei der Interaktion verschiedenener Systemkomponenten erreicht, indem die Camel API angesprochen wird, ohne auf konkrete Protokolle und Datentypen zuzugreifen.

Über diesen Mechanismus können sogenannte Endpunkte veröffentlicht werden. Endpunkte sind Schnittstellen, die den Zugriff auf einzelne Systemkomponenten als Servicedefinitionen realisieren. Diese können sowohl für den internen als auch externen Zugriff bereitgestellt werden. Mittels der Integrationsplattform Apache CXF³ werden auf diese

¹<http://karaf.apache.org/> [Online; Letzter Zugriff: 30.06.2013]

²<http://camel.apache.org/> [Online; Letzter Zugriff: 30.06.2013]

³<http://cxf.apache.org/> [Online; Letzter Zugriff: 30.06.2013]

Weise Web-Services per contract-first-Ansatz generiert und in die Camel Routen als Endpunkte eingehängt. Apache CXF unterstützt eine Vielzahl an Implementierungen und standardisierten Kommunikationsprotokollen, wobei im Rahmen des Beschaffungsprojektes vor allem SOAP Web-Services generiert werden. Andere Technologien und Protokolle könnten ebenfalls umgesetzt werden, um beispielsweise Legacy-Systeme zu integrieren. Dadurch wird eine prozessorientierte Sichtweise realisiert.

Die Geschäftslogik für die eigentlichen Beschaffungsprozesse wird über Activiti⁴ realisiert. Activiti [Ra12] ist ein quelloffenes Framework für die Modellierung nach BPMN 2.0 sowie die Implementierung und die Ausführung von Geschäftsprozessen. Es wird ebenfalls in die Routing-Plattform eingehängt.

Die Persistierung der Daten erfolgt mittels OpenJPA⁵, bei welchem es sich um eine Java EE 5 Spezifikation handelt, die aus Hibernate und anderen ORM-Technologien hervorgegangen ist. Die Persistenzschicht greift auf eine PostgreSQL-Datenbank zu, wobei das System nicht auf diese beschränkt ist.

Des Weiteren muss das Beschaffungssystem mit dem MBS Modul in Version 11.0 des Hochschulinformationssystems (HIS) kommunizieren. Zu diesem Zweck wird die durch das Modul bereitgestellte Dateischnittstelle zur Übergabe von Buchungsdaten an HISFSV-GX verwendet, welche aufgrund der Anforderungen seitens des Rechenzentrums der Hochschule ausschließlich genutzt werden soll. Auf diese Weise wird der Export und Import von Buchungsdaten zwischen HIS-Systemen und der Import in HIS-Systeme aus Fremdsystemen realisiert. Auf Basis der Schnittstellendokumentation wurde eine Bibliothek entwickelt, die den Import und die Generierung entsprechender HIS-Dateien über das Beschaffungssystem erlaubt. HIS-Dateien für den Import in das Beschaffungssystem können somit in einem ausgewiesenen Ordner abgelegt werden, von wo aus diese automatisch verarbeitet werden. Diese Verarbeitung wird mittels Apache Camel realisiert. Weiterhin ist das Beschaffungssystem in der Lage, Beschaffungsanträge in das HIS-Format für den HIS-Import zu konvertieren. Dies geschieht ebenfalls automatisiert.

Architektur

Nach Abschluss des studentischen Projektes wird dieses an das Rechenzentrum der Universität übergeben, welches fortan die Plattform eigenständig warten und weiterentwickeln wird. Aus diesem Grund hängen der erfolgreiche Einsatz sowie die Wartbarkeit und Weiterentwicklung des Beschaffungssystems vor allem davon ab, wie gut weitere Entwickler dieses System aufnehmen und die inhärente Lernkurve bewältigen. Deshalb stehen die Aspekte Transparenz, Auffindbarkeit und Modularisierung im Fokus des Entwurfs der Systemarchitektur.

Um größtmögliche Transparenz zu erzielen, erfolgt nicht nur eine strikte Trennung von Verantwortungsbereichen durch Modularisierung, sondern es wird auch darauf Wert gelegt, den Anteil an Verbindungscode auf ein Minimum zu reduzieren. Auf diese Weise wird für projektfremde Entwickler eine nachvollziehbare Strukturierung angestrebt.

⁴<http://www.activiti.org/> [Online; Letzter Zugriff: 30.06.2013]

⁵<http://openjpa.apache.org/> [Online; Letzter Zugriff: 30.06.2013]

Der Aspekt der Auffindbarkeit zielt darauf ab, wie hoch die kognitive Hürde zum Verstehen des Kontrollflusses für den Entwickler ist. Aus diesem Grund wird strikt zwischen dem Modell und dessen Implementierung unterschieden. Darüber hinaus sind Schnittstellen orthogonal konzipiert und es werden flache Hierarchien für Methodenaufrufe angestrebt.

Unter Berücksichtigung dieser Überlegungen wurde eine dreischichtige Systemarchitektur nach dem bottom-up-Prinzip entworfen. Diese ist in Abbildung 2 dargestellt.

Herzstück des Beschaffungssystems - die Laufzeitumgebung - bildet die OSGi-Plattform, welche über Apache Karaf bereitgestellt wird. Die Veröffentlichung neuer Funktionalitäten erfolgt in Form von Bundels. Hierbei handelt es sich um eine Sammlung von Java-Paketen, denen über eine sogenannte *MANIFEST*-Datei explizit Informationen für den *Classpath* mitgegeben werden. Unter anderem wird festgelegt, welche Schnittstellen exportiert werden, sodass andere diese einbinden können. Auf diese Weise erfolgt die Bundle-übergreifende Kommunikation ausschließlich anhand der Schnittstelleninformationen.

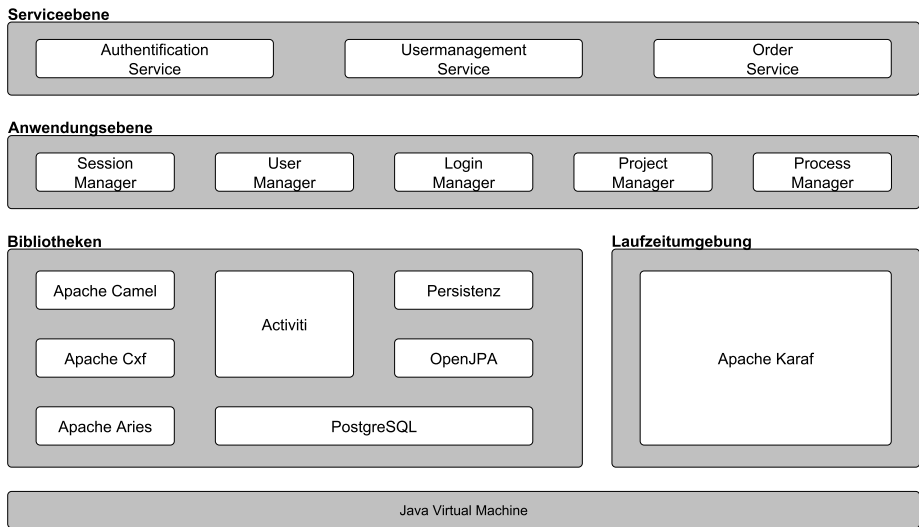


Figure 2: Architektur des entwickelten Beschaffungssystems

Sowohl die Service- als auch die Anwendungsebene greifen auf eine Menge von Basisbibliotheken zu, die als OSGi-Bundles auf der Plattform bereitgestellt werden. Hierunter fallen diverse quelloffene Bibliotheken für die Ansteuerung von Datenbanken, die Persistierung von Datensätzen, die Ansteuerung der Prozessengine oder auch die automatisierte Instanziierung von Klassen via *Dependency Injection* usw.

Auf der Anwendungsebene werden Funktionen in Form von Domänenobjekten aggregiert, so dass diese auf einer abstrakteren Ebene wiederverwendet werden können. Auf diese Weise werden verschiedene Programmierschnittstellen bereitgestellt, die eine Kommunikation zwischen einzelnen Anwendungen und der Datenschicht realisiert.

Die Webservices befinden sich auf der Serviceebene. Sie setzen die Geschäftslogik um und greifen dabei auf Komponenten der Anwendungsebene zu. Über die Serviceebene erfolgt die Kommunikation mit dem Frontend.

3.2 Frontend

Die grafische Benutzungsschnittstelle (GUI) stellt für den Nutzer des System eine besonders wichtige Komponente dar. Um einen Einblick in das Konzept zu geben, werden in diesem Abschnitt technologische Grundlagen und Überlegungen vorgestellt und anschließend ein Einblick in das Design der GUI gegeben.

Technologien

Zuerst werden die technologischen Aspekte der GUI dargelegt. Dazu werden im Folgenden einige grundlegende Überlegungen geschildert. Nachstehend wird spezielles Augenmerk auf das der GUI zu Grunde liegende Zend-Framework⁶ gelegt.

Bereits zu Beginn der GUI-Implementierung wurde entschieden auf Technologien wie XHTML⁷, CSS⁸ und PHP⁹ zu setzen, um zukunftsfähig zu sein und einer großen Masse an potentiellen Nutzern mit Hilfe eines beliebigen Webbrowsers leichten Zugang zum System zu ermöglichen. Dabei sollen vorrangig die gängigen Desktop-Browser unterstützt werden [Re13]: Chrome, Firefox, Internet Explorer, Opera und Safari. Kleinere Probleme hat in diesem Zusammenhang die geplante Darstellung der Beschaffungsformulare als XForms¹⁰ gemacht, da nach Kenntnisstand keiner der gängigen Browser die XForms Spezifikation¹¹ implementiert hat. Gelöst wurde diese Problematik mit konsequenter Ablösung der XForms durch HTML¹² Version 5.

Für die Wahl eines geeigneten Frameworks für die Implementierung des Frontends ist es wichtig, dass Frameworks nach unterschiedlichen Kriterien differenziert werden können. So besteht eine Unterscheidung zwischen Frameworks für eine direkte beziehungsweise eine externe Anbindung. Weiterhin existieren seiten- und anwendungsorientierte Frameworks.

Für die Implementierung des Frontends mit dem Vaadin Framework¹³ sprach zunächst die Möglichkeit, Vaadin als OSGi-Modul direkt an das Backend zu koppeln. Der Hauptvorteil dieser Lösung liegt in dem für das Frontend sehr einfachen Zugang zu den Funktionalitäten des Backends. Gegen diesen Ansatz sprach das SOA-Entwurfsprinzip der losen Kopplung. Das Frontend sollte möglichst unabhängig vom Backend implementiert wer-

⁶<http://framework.zend.com/> [Online; Letzter Zugriff: 30.06.2013]

⁷<http://www.w3.org/html/> [Online; Letzter Zugriff: 30.06.2013]

⁸<http://www.w3.org/TR/CSS2/> [Online; Letzter Zugriff: 30.06.2013]

⁹<http://php.net/> [Online; Letzter Zugriff: 30.06.2013]

¹⁰<http://www.w3.org/TR/2009/REC-xforms-20091020/> [Online; Letzter Zugriff: 30.06.2013]

¹¹<http://php.net/manual/de/features.xforms.php> [Online; Letzter Zugriff: 30.06.2013]

¹²<http://www.w3.org/TR/html5/> [Online; Letzter Zugriff: 01.07.2013]

¹³<https://vaadin.com/> [Online; Letzter Zugriff: 30.06.2013]

den. Sinnvoller schien in diesem Kontext die Implementierung des Frontends auf Basis einer externen Anbindung mit Hilfe von Web-Services.

Seiten- beziehungsweise anwendungsorientierte Frameworks können aufgrund ihres Umgangs mit den dargestellten Inhalten voneinander abgegrenzt werden. Seitenorientierte Frameworks rendern jeweils die komplette Darstellung, wenn der Benutzer größere Aktionen durchführt. Anwendungsorientierte Frameworks laden hingegen zur Laufzeit per AJAX Inhalte nach. Dies führt zu starker Abhängigkeit und zieht unter anderem das Problem nach sich, dass der Benutzer nur schwer ein Lesezeichen auf einen konkreten Zustand des Systems setzen kann. Ein Kandidat für die Frontend-Implementierung war deshalb das mittlerweile in Version 2 vorliegende Zend-Framework, da dieses aufgrund seiner Seitenorientierung praktisch erschien.

Letztlich fiel die Wahl auf das PHP-basierte Zend-Framework auch deshalb, da die Implementierung des Frontends auf Basis einer externen Anbindung der Sicherheit des Backends und dem Prinzip der losen Kopplung zu Gute kommt. Für die Nutzung des Zend-Frameworks spricht zudem der ihm zu Grunde liegende Model-View-Controller-Ansatz [GHJ01], welcher eine strikte Trennung der einzelnen Frontend-Komponenten ermöglicht. Das Zend-eigene individuell anpassbare Routing und die Möglichkeit, CSS im Zusammenhang mit PHP für die Gestaltung der einzelnen Präsentationen zu nutzen, fördern eine zukünftige einfache Wartbarkeit. Auf Grund der Seitenorientierung des Zend-Frameworks lassen sich darüber hinaus auf unproblematische Art und Weise Mittel zur Sicherstellung der Browser-übergreifenden Kompatibilität umsetzen. So sollen alle Präsentationen sowohl mit Javascript als auch ohne Javascript angezeigt und genutzt werden können.

Insgesamt orientierte sich die Umsetzung des Frontends an unterschiedlichen, minimalistischen Prinzipien. So wird so wenig Code wie möglich verwendet, aber so viel wie nötig, um die volle Funktionalität für den Nutzer zu gewährleisten. Auch das Model-View-Controller-Prinzip wurde mit der Kombination des Zend Frameworks und HTML5 konsequent umgesetzt. Zudem wurde darauf geachtet, keinerlei Backend-Logik im Frontend zu implementieren und zugleich die Schnittstelle zum Backend so simpel wie möglich gestalten. Die mit Hilfe von HTML5 modellierten Beschaffungsformulare wurden in dynamische Layouts integriert, um den Anpassungsaufwand bei Layoutveränderungen so gering wie möglich zu halten.

Design

Im weiteren Verlauf wird der für die GUI wichtige Design-Prozess erläutert, damit schließlich ein Einblick in einen Design-Prototypen gewährt werden kann. Neben der Anpassung der GUI an das Cooperate Design der Universität Potsdam wurde sehr viel Wert auf Nutzernähe und -verträglichkeit gelegt. Dafür wurde eine umfangreiche, das ganze Projekt hinweg andauernde, nutzerorientierte Bedarfsanalyse im Gespräch mit Mitarbeitern verschiedener Bereiche durchgeführt, wie von Foraker Labs [Fo13] vorgeschlagen.

Die Bedarfsanalyse hat hinsichtlich der GUI aufgezeigt, dass sich die späteren, nicht aus der IT stammenden Benutzer eine klare, übersichtliche und eingängliche Darstellung der gezeigten Inhalte wünschen. Besonderes Augenmerk legen die Befragten auf eine

prägnante Präsentation der Beschaffungsformulare. Diese sollen den bereits existierenden Papierformularen weitestgehend nachempfunden werden, sodass eine Umgewöhnung im Rahmen der Umstellung auf digitale Formulare wegfällt beziehungsweise so minimal wie möglich ausfällt. Die späteren Nutzer haben sich im Verlauf der Bedarfsanalyse stets rückversichert, ob verschiedene Zusatzfunktionen, die nur Papier bietet, digital zu realisieren sind. Explizit wurde zum Beispiel nach einer Möglichkeit gesucht, Aktennotizen darzustellen. Ein weiteres wichtiges Designmerkmal, welches sich aus den Nutzerbefragungen ergeben hat, ist das konsequente Implementieren von flachen Strukturen. Zum Beispiel werden Tabellen mit Registerkarten und nicht mit Menüs behaftet dargestellt. Diese lockern das Design auf und fördern die Übersicht aus Nutzerperspektive. Das später gezeigte Enddesign orientiert sich stark an den im Kontext der Bedarfsanalyse erhobenen Anforderungen.

Die designtechnische Umsetzung begann bereits in sehr frühen Phasen der Befragung der späteren Nutzer; so wurden schon als Ergebnis der ersten Gespräche Design Mockups [Fo13] kreiert und dazu Feedback von den Nutzern erfragt. In späteren Designzyklen folgten detaillierte Grafiken, statische und schließlich interaktive Prototypen. Abbildung 3 zeigt einen Ausschnitt aus dem dann verfügbaren Online-Beschaffungssystem.

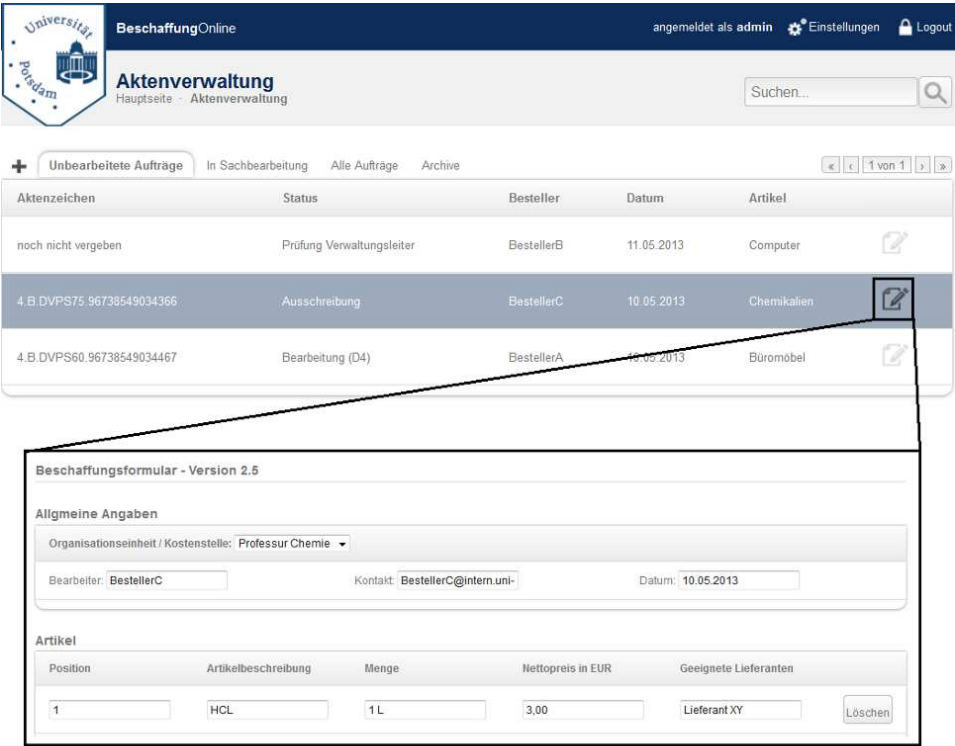


Figure 3: Benutzeroberfläche des Online Beschaffungssystems

Am gezeigten Ausschnitt des Systems sind untergeordnete Designaspekte deutlich sichtbar, wie die Verwendung serifenfreier Schriften für die bessere Lesbarkeit [Ha08] sowie eine weitestgehend kontrastreiche Farbgestaltung, die neben klarer Übersicht ebenfalls für eine bessere Lesbarkeit der dargestellten Inhalte sorgt.

4 Zusammenfassung & Ausblick

Das entstandene System setzt den erfassten Beschaffungsprozess über eine Web-basierte Schnittstelle um. Dadurch werden Übertragungsfehler vermieden und Bearbeitungszeiten verkürzt. Der Fortschritt der Bearbeitung wird beständig für alle Beteiligten transparent gemacht. Zusätzlich können Automatismen zum Einsatz kommen, wie etwa bei der Auswahl von Kostenstellen oder bei der Bearbeitung in Abhängigkeit erreichter Wertgrenzen. Das System wird derzeit von einem Kernteam überarbeitet und in den Produktivbetrieb überführt.

Dafür sind noch letzte Designanpassungen, ein breiter Browserkompatibilitätstest sowie die abschließende Evaluation gemeinsam mit verschiedenen Nutzergruppen erforderlich. Das System soll dann in einem gestuften Einführungsprozess bis zum Jahresende 2013 vollumfänglich in Betrieb genommen werden.

Noch nicht enthalten sind in der derzeitigen Systemkonzeption weiterführende Funktionen wie z.B. Kataloge ausgewählter Anbieter oder die Anbindung an den Vergabemarktplatz des Landes. Derartige Komponenten können in späteren Schritten folgen.

Danksagung

Wir danken den weiteren im WS 2011/12 und im SS2012 am Projekt beteiligten Studierenden Tommy Ball, Jan Brennenstuhl, Nils Dressel, Kora Haedge, Thomas Hein, Christoph Tobias König, Martin Mattern, Jonas Rogge, Vyacheslav Rogozhkin, Moritz Rumpf, Paul Schröder, Max Stahl, Heiko Thiel, Christian Wegner und Volker Wölfert für ihr großartiges Engagement sowie den beteiligten Mitarbeitern in Technik und Verwaltung für die angenehme und konstruktive Zusammenarbeit.

References

- [AM10] Altenschmidt, K. ; Miller, J. : Service Learning in der Hochschuldidaktik. Budrich Uni-Press, Opladen, 2010.
- [co13] cosinex GmbH: cosinex GmbH - Lösungen im Bereich E-Government. <http://www.cosinex.de/index.html>, 2013. [Online; Letzter Zugriff: 30.06.2013].
- [CW13] Cummins, H. ; Ward, T. : Enterprise Osgi in Action. With Examples Using Apache Aries. Manning Publications Co., Greenwich, 2013.

- [Fo13] Foraker Labs: Usability First - About Usability - Website Design. <http://www.usabilityfirst.com/about-usability/website-design/>, 2013. [Online; Letzter Zugriff: 30.06.2013].
- [GHJ01] Gamma, E. ; Helm, R. ; Johnson, R. : Entwurfsmuster. Elemente wiederverwendbarer objektorientierter Software. Addison-Wesley, Bonn, 2001.
- [GI13] GISA GmbH: GISA GmbH. <http://www.gisa.de/index.html>, 2013. [Online; Letzter Zugriff: 30.06.2013].
- [Ha08] Hanyaloglu, A. : To Serif or Not To Serif? Regarding Online Readability. http://blogs.adobe.com/acom/2008/07/to_serif_or_not_to_serif_regar_1.html, 2008. [Online; Letzter Zugriff: 30.06.2013].
- [He12] Herrmann, T. : Kreatives Prozessdesign. Konzepte und Methoden zur Integration von Prozessorganisation, Technik und Arbeitsgestaltung. Springer, Berlin, 2012.
- [IAZ11] Ibsen, C. ; Anstey, J. ; Zbarcea, H. : Camel in Action. Manning Publications Co., Greenwich, 2011.
- [Ra12] Rademakers, T. : Activiti in Action: Executable Business Processes in Bpmn 2.0. Manning Publications Co., Greenwich, 2012.
- [Re13] Refsnes Data: Browser Statistics. http://www.w3schools.com/browsers/browsers_stats.asp, 2013. [Online; Letzter Zugriff: 30.06.2013].
- [Rö13] Rödl & Partner: IT-Lösungen — Rödl & Partner. <http://www.roedl.de/dienstleistungen/unternehmen-it-beratung/itloesungen/>, 2013. [Online; Letzter Zugriff: 30.06.2013].
- [ZKB12] Zakhariya, H. ; Kosch, L. ; Breitner, M. H. : Elektronische Drittmittelakte in der Hochschulverwaltung - Erkenntnisse aus Fallstudien. In Glück, H. I. (Hrsg.): Proc. 42. Jahrestagung der Gesellschaft für Informatik, LNI P-208, S. 613–626. Köllen, Bonn, 2012.