



OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

HW

FAKULTÄT FÜR
HUMANWISSENSCHAFTEN

**Henry Frankenberger:
Informationstechnologien in der
Berufsbildungsforschung. Eine experimentelle
Untersuchung über Aufbau und Anwendung
computergestützter Analysemethoden in der
Berufsbildung.**

Hrsg. von Prof. Dr. Dietmar Frommberger
Heft 2 | 2013
ISSN 1865-2247

Herausgeber:

Prof. Dr. Dietmar Frommberger

Lehrstuhl für Berufspädagogik

Fakultät für Humanwissenschaften

Institut für Berufs- und Betriebspädagogik

Otto-von-Guericke-Universität Magdeburg

Zschokkestraße 32

D-39104 Magdeburg

Telefon: +49-(0)391-67-56 625

Telefax: +49-(0)391-67-56 550

E-Mail: dietmar.frommberger@ovgu.de

Quelle/ Zitationshinweis:

Frankenberger, H. (2013): Informationstechnologien in der Berufsbildungsforschung. Eine experimentelle Untersuchung über Aufbau und Anwendung computergestützter Analysemethoden in der Berufsbildung.. In: Frommberger, D. (Hrsg.), *Magdeburger Schriften zur Berufs- und Wirtschaftspädagogik*, Heft 2, Jg. 2013. Otto-von-Guericke-Universität Magdeburg.

Online Zugriff unter:

http://www.ibbp.ovgu.de/Institut/Berufsp%C3%A4dagogik/Magdeburger+Schriften+zur+Berufs_+und+Wirtschaftsp%C3%A4dagogik/Jahrgang+2013.html

© Copyright

Die in der Reihe *Magdeburger Schriften zur Berufs- und Wirtschaftspädagogik* erscheinenden Veröffentlichungen sind einschließlich Graphiken und Tabellen urheberrechtlich geschützt. Jede Verwendung außerhalb der Grenzen des Urheberrechts ist ohne Zustimmung des Herausgebers unzulässig. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Microverfilmungen und Einspeicherung auf elektronischen Datenträger.

Abstract

Schon im späten Mittelalter zur Zeit der Gilden, Zünfte und Innungen ergab sich in Mitteleuropa die Notwendigkeit, berufsbezogene Daten zu erfassen, um die Produktivität der merkantilen Staaten zu optimieren. Dabei hatten die berufsständischen Körperschaften nicht nur die Aufgabe, Ausbildungszeit, -verlauf und Lehrgeld zu ordnen, sie waren auch verpflichtet, die Anzahl der Zu- und Abgänge innerhalb der einzelnen Berufsbereiche zu überwachen, um ein Ungleichgewicht an ausgebildeten Kräften zu vermeiden: Denn dies hätte die wirtschaftliche und soziale Basis der ständisch geordneten Gesellschaft gefährden können (vgl. Althoff 2006, S. 199-200).

Auch heute noch sind berufsbezogene Daten ein unverzichtbarer Bestandteil unserer gesellschaftlichen Entwicklung. Sie werden in der Regel durch die Kammern der Wirtschaft erfasst und an die Statistischen Landesämter weitergeleitet, bis sie schließlich dem Bundesinstitut für Berufsbildungsforschung zur Verfügung gestellt werden. Dort werden die Daten von einer separaten Organisationseinheit zur Berufsbildungsforschung verwendet. Darüber hinaus obliegt es dieser Organisationseinheit, den Berufsbildungsbericht zu erstellen (vgl. § 88 Abs. 4 BBiG).

Dank moderner Technik liegen die Datensätze, Informationen und Berichte heute meist in elektronischer Form vor. Dennoch haben bislang nur wenige Menschen Zugriff auf Primärdaten und Urlisten. Eine übergreifende Nutzung, die wissenschaftlich tätige Personen, Hochschuleinrichtungen und anderen Interessengruppen gleichermaßen den Zugriff erlaubt und überdies die Partizipation an Forschungsergebnissen und -projekten gewährleistet, ist bisher in der Berufsbildungsforschung noch nicht umgesetzt worden. Dabei könnte die Verbreitung von forschungsrelevanten Daten einen hohen Mehrwert für die Berufsbildungsforschung bedeuten. Nicht nur die zuständigen Stellen sind dabei von Bedeutung, sondern jeder, der ein gewisses Interesse daran hegt, berufsbezogene Daten für Forschungszwecke zu verwenden oder zu verbreiten. Zu fragen wäre folglich: Wie kann es gelingen, die vorliegenden berufsbezogenen Daten effizienter zu nutzen als bisher und wie müsste die informationstechnologische Infrastruktur der Berufsbildungsforschung beschaffen sein, um forschungsrelevante Daten übergreifend verwenden zu können? Diese Fragen sollen im Rahmen der vorliegenden Forschungsarbeit beantwortet werden.

Die Arbeit gliedert sich in fünf Teile. Zunächst werden aktuelle Informationssysteme im Bereich der Berufsbildungsforschung analysiert. Danach folgt eine kritische Betrachtung dieser Systeme. Auf dieser Grundlage wird schließlich das Ziel der Arbeit beschrieben: Es

besteht in der Entwicklung eines Konzeptes, das eine Mängelminimierung hinsichtlich der dargestellten Defizite vorsieht. Schließlich wird dieses Konzept unter modellhaften Annahmen simuliert. Abschließend werden die Ergebnisse der Simulation ausgewertet und auf die Forschungsfragen bezogen diskutiert.

Henry Frankenberger: Informationstechnologien in der Berufsbildungsforschung. Eine experimentelle Untersuchung über Aufbau und Anwendung computergestützter Analysemethoden in der Berufsbildung.

Inhaltsübersicht

| | | |
|-------|---|----|
| 1 | Ausgangssituation | 3 |
| 2 | Problemdarstellung und Forschungsziele | 6 |
| 3 | Abgrenzung des Forschungsgegenstandes | 8 |
| 3.1 | Vorbemerkung zur Abgrenzung | 8 |
| 3.2 | Abgrenzung des informationstechnologischen Forschungsgegenstandes | 9 |
| 3.3 | Abgrenzung des berufspädagogischen Forschungsgegenstandes..... | 12 |
| 4 | Methoden..... | 16 |
| 4.1 | Vorbemerkung zur Methodik | 16 |
| 4.2 | Migrationskonzept..... | 17 |
| 4.2.1 | Vorgehensweise..... | 17 |
| 4.2.2 | Analyse | 25 |
| 4.2.3 | Entwurf..... | 29 |
| 4.2.4 | Realisierung..... | 49 |
| 4.3 | Durchführung des Experiments..... | 52 |
| 4.3.1 | Vorbemerkung zum Experiment | 52 |
| 4.3.2 | Vorbereitung..... | 53 |
| 4.3.3 | Simulationsbeschreibung..... | 54 |
| 4.3.4 | Ziele der Simulation | 54 |
| 4.3.5 | Testverfahren und -protokoll..... | 55 |
| 4.3.6 | Testergebnisse | 56 |
| 4.3.7 | Auswertung | 61 |
| 5 | Schlussfolgerung | 63 |
| 6 | Empfehlungen für weiterführende Forschungsansätze..... | 66 |
| | Literatur..... | 67 |

| | |
|-----------------------------------|----|
| Anlagen | 70 |
| Datentypen | 70 |
| Serverkonfiguration..... | 75 |
| Zusammengefasste WSDL-Datei | 76 |

Abbildungsverzeichnis

| | |
|---|----|
| <i>Abbildung 1:</i> Genutzte computergestützte Analysesoftware | 5 |
| <i>Abbildung 2:</i> Erhebungsmerkmale im Kontext der Berufsbildungsstatistik. Entnommen aus dem Berufsbildungsgesetz. | 14 |
| <i>Abbildung 3:</i> Kernelemente einer SOA nach Seth, Agarwal und Raj Singla (2011), erweitert um die funktionalen Beziehungen zwischen den einzelnen Elementen auf der Grundlage von Schill und Springer (2012, S. 20)..... | 31 |
| <i>Abbildung 4:</i> Webservice-Technologien im Überblick nach Dumke et al. (2003, S. 154)..... | 32 |
| <i>Abbildung 5:</i> Enterprise-Service-Bus der serviceorientierten Forschungsdateninfrastruktur.. | 34 |
| <i>Abbildung 6:</i> Semantisches Datenmodell, dargestellt durch das Entity-Relationship-Diagramm..... | 39 |
| <i>Abbildung 7:</i> Das relationale Datenmodell zur persistenten Datenspeicherung. | 40 |
| <i>Abbildung 8:</i> Die standardisierten Datentypen der Dienstanbieter. | 42 |
| <i>Abbildung 9:</i> Grobentwurf einer serviceorientierte Forschungsdateninfrastruktur am Beispiel der Lehrstellenzählung. Erstellt auf der Grundlage der Geschäftsprozessanalyse; Services arbeiten choreografisch zusammen. | 46 |
| <i>Abbildung 10:</i> Feinentwurf einer serviceorientierte Forschungsdateninfrastruktur am Beispiel der Lehrstellenzählung und -analyse..... | 48 |
| <i>Abbildung 11:</i> Die Adresse der zentralen Verwaltungseinheit RES auf dem Zielsystem. | 50 |
| <i>Abbildung 12:</i> Die WSDL-Adresse eines Webservice..... | 50 |
| <i>Abbildung 13:</i> Eine Möglichkeit zum Importieren von Webservices in Java..... | 50 |
| <i>Abbildung 14:</i> Benutzeroberfläche der zentralen Verwaltungseinheit RES mit Service-Portfolio zur Lehrstellenzählung..... | 51 |

Tabellenverzeichnis

| | |
|---|----|
| <i>Tabelle 1:</i> Gesamtprozess der Lehrstellenzählung als Grundlage für die Entwicklung von Services. | 28 |
| <i>Tabelle 2:</i> Identifizierung der Entitäten zur Entwicklung von standardisierten Datentypen... 38 | |
| <i>Tabelle 3:</i> Der Aufbau des Ereignisprotokolls..... | 55 |
| <i>Tabelle 4:</i> Das Ergebnisprotokoll des 1. Testdurchlaufes..... | 56 |
| <i>Tabelle 5:</i> Das Ergebnisprotokoll des 2. Testdurchlaufes..... | 59 |

1 Ausgangssituation

Eines der wichtigsten Analyse- und Informationsinstrumente der Berufsbildungsforschung ist der Berufsbildungsbericht. Er beinhaltet Informationen über den Stand und die Weiterentwicklung der Berufsbildung in Deutschland und ist somit nicht nur für wissenschaftlich tätige Personen und Einrichtungen von Bedeutung, sondern auch für die Bundesregierung und die Sozialpartner. Für die Erstellung des Berufsbildungsberichtes ist das Bundesinstitut für Berufsbildungsforschung zuständig. Dieses hat laut § 90 Abs. 2 Berufsbildungsgesetz (BBiG) die Aufgabe, die Ergebnisse ihrer Forschungsleistungen zu veröffentlichen. Aus diesem Grund wird der jährliche Berufsbildungsbericht auf den Internetseiten des BIBB in Form eines plattformunabhängigen Dateiformates (PDF) für wissenschaftlich tätige Personen und Einrichtungen veröffentlicht. Das Dokument selbst gliedert sich in zwei Teile: Während der erste Teil eine Stellungnahme der Bundesregierung zur aktuellen Situation auf dem Ausbildungsmarkt sowie die Darstellung der Regierungspolitik zur Lösung aktueller Probleme umfasst, enthält der zweite Teil Berichte und Analysen über Datengrundlagen, Statistiken, Forschungsprojekte und aktuelle Entwicklungen in der beruflichen Aus- und Weiterbildung. Beiden Teilen ist jedoch gemein, dass Informationen in textueller Form präsentiert werden (vgl. Krekel 2006, S. 204-206).

Die Anfertigung des Berufsbildungsberichtes verlangt laut § 88 Abs. 4 BBiG die Übermittlung der statistischen Daten an das BIBB. Dies setzt wiederum die Erhebung und Bereitstellung von Datensätzen voraus. Im Bereich der dualen Berufsausbildung werden dazu zwei verschiedene Datenerhebungen bei den für die Berufsausbildung zuständigen Stellen durchgeführt: zum einen die vom BIBB angeordnete Erhebung der neu abgeschlossenen Ausbildungsverträge und zum anderen die Berufsbildungsstatistik der statistischen Ämter des

Bundes und der Länder. Während die statistischen Ämter des Bundes und der Länder seit 1977 die Berufsbildungsstatistik zum 31.12. eines jeden Jahres erheben, führt das BIBB seit 1976 eine jährliche Zählung der Ausbildungsverträge durch, die im Zeitraum vom 01. Oktober des Vorjahres bis zum 30. September des Erhebungsjahres neu abgeschlossen wurden und am 30. September noch bestanden. Die Erhebungen des BIBB basieren auf der Grundlage des § 86 BBiG, wohingegen die der statistischen Ämter aus § 88 BBiG hervorgehen. Beide Erhebungsformen beginnen allerdings bei den zuständigen Stellen. Diese übermitteln zum einen die Anzahl der neu abgeschlossenen Ausbildungsverträge an das BIBB und zum anderen die von ihnen erhobenen Individualdaten mit unterschiedlicher Merkmalsausprägung an die Statistischen Landesämter sowie an das Statistische Bundesamt. In beiden Fällen werden die Daten auf dem elektronischen Weg übermittelt. Im Bereich der Berufsbildungsstatistik, die zum 31.12. von den Statistischen Landesämtern und dem Bundesamt für Statistik durchgeführt wird, stehen die Lieferwege eSTATISTIK.core sowie IDEV zur Verfügung. Beide Varianten bieten einen Zugang zu den Datensätzen auf Online-Portalen an. (vgl. Uhly et al. 2009, S. 2)

Eine in der Fachliteratur der Berufsbildungsforschung bisher noch wenig diskutierte Datenerfassungsmethode stellt das 2008 gegründete Forschungsdatenzentrum des BIBB (FDZ-BIBB) dar. Dabei handelt es sich um eine elektronische Forschungsdateninfrastruktur, mithilfe derer wissenschaftlich tätige Personen und Einrichtungen unter Einbeziehung computergestützter Analysesoftware an den quantitativen Datenbeständen der BIBB-Forschungsprojekte partizipieren können. In einer Online-Umfrage untersuchten Friedrich, Hohn und Rohrbach-Schmidt (2011) die Nutzung und die Akzeptanz der FDZ-BIBB vonseiten wissenschaftlich tätiger Personen. Dabei gingen die Autoren zunächst der Frage nach, welche computergestützte Analysesoftware von den 101 befragten Personen am häufigsten zur Auswertung von quantitativen Daten eingesetzt werde. Die Ergebnisse der Untersuchung zeigen, dass die Produkte SPSS und STATA die gängigsten Programme sind. Auch die FDZ-BIBB biete laut Friedrich, Hohn und Rohrbach-Schmidt hauptsächlich ihre Datenbestände in Formaten an, die sich innerhalb dieser beiden Softwareprodukte integrieren lassen.

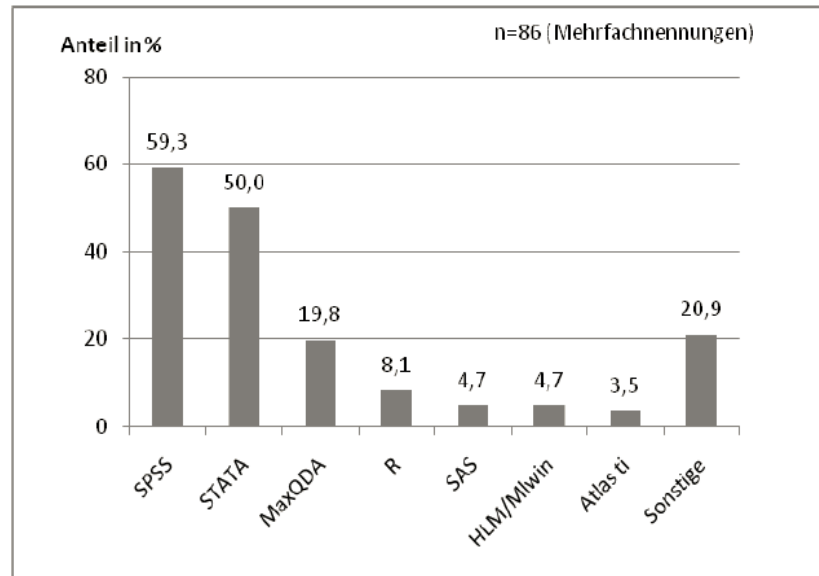


Abbildung 1: Genutzte computergestützte Analysesoftware nach Friedrich, Hohn und Rohrbach-Schmidt (2011, S. 6).

Dennoch belegen die Studien von Friedrich, Hohn und Schmidt (2011, S. 10), dass nur rund ein Drittel (36,3 Prozent) der Befragten bereits einmal mit Daten des BIBB-FDZ gearbeitet hat (vgl. ebd., S. 10). Von diesen Personen wiederum stammen lediglich 14 Prozent aus dem Bereich der Berufs- und Wirtschaftspädagogik. Diese Tatsache gab den Autoren Anlass, die Teilnehmer um eine Beurteilung des Angebotes der BIBB-FDZ zu bitten. Jene konnten ihre Bewertung auf einer Skala von 1 („sehr gut“) sowie 4 („sehr schlecht“) angeben. Aus den Untersuchungen lässt sich insgesamt ein positives Bild ablesen: Das Angebot an Personendaten wird dabei mit einem Mittelwert von $x=1,7$ am besten bewertet. Allerdings werden die Informationen zum Datenzugang von den Datennutzern nur als gut ($x=2,0$) eingeschätzt.

Mit der Implementierung der BIBB-FDZ wurde 2008 eine technologische Grundlage geschaffen, die sich auch auf andere Gebiete ausweiten könnte. So hatte beispielsweise der Europäische Rat (2000) in Lissabon die Schaffung eines gemeinsamen Forschungsraumes in Europa gefordert. Dabei sollten die Mitgliedstaaten die erforderlichen Schritte unternehmen, um

„geeignete Mechanismen für die Vernetzung von nationalen und gemeinsamen Forschungsprogrammen auf freiwilliger Grundlage im Rahmen frei gewählter Ziele zu entwickeln, damit die abgestimmten Ressourcen für die F&E in den Mitgliedstaaten besser genutzt werden und eine regelmäßige Berichterstattung an den Rat über die erzielten Fortschritte sichergestellt wird“ (Europäischer Rat 2000).

Dieses Ziel wurde 2007 in Brüssel novelliert. Dort hob der Rat die Notwendigkeit hervor, Synergien zwischen Gemeinschaftsprogrammen zu optimieren, um die Umsetzung von Forschungsergebnissen in innovative Produkte und Dienstleistungen zu verbessern. Den Bestimmungen des Rates zufolge haben die Mitgliedstaaten sicherzustellen, dass Wissen in angemessener Weise an alle Partner weitergegeben wird (Europäischer Rat 2007).

Die Vernetzung von Forschungsprogrammen und die Verbreitung von berufsbezogenen Datenbeständen nach dem Vorbild der BIBB-FDZ ist deshalb auch ein Thema, das die internationale Berufsbildungsforschung in Zukunft vermehrt betreffen könnte. Schließlich ersucht der Rat bezogen auf die Strategie „Europa 2020“ derzeit nach neuen Möglichkeiten, um „eine gründliche Analyse der Fortschritte im Hinblick auf die Kernziele und die ‚ET 2020‘-Referenzwerte vorzulegen, die als Grundlage für einen Gedankenaustausch im Rat während jedes ‚Europäischen Semesters‘ dienen soll“ (Europäischer Rat 2011).

2 Problemdarstellung und Forschungsziele

Die bisherigen Ausführungen zeigen, dass berufsbezogene Daten und Analysen in sehr unterschiedlicher Form vorliegen: entweder in elektronischen Dokumenten nach dem Vorbild des Berufsbildungsberichtes, in elektronischen Datenbanken, wie sie von den Statistischen Landesämtern zur Verfügung gestellt werden, oder in speziellen Datenformaten des BIBB-FDZ. Den ersten beiden Formen ist gemeinsam, dass Daten in geschlossenen Strukturen organisiert werden (vgl. Althoff 2006, S. 202). So kann z. B. der Datenreport zum Berufsbildungsbericht zwar eingesehen, aber für weitere sekundärstatistische Untersuchungen nur sehr umständlich verwendet werden, weil das zugrundeliegende Dokumentenformat den Zugriff von außen verweigert. Auch die Datenbestände in den Datenbanken der statistischen Ämter unterliegen einer begrenzten Wiederverwendung (vgl. ebd. S. 204). So können diese von unterschiedlichen Interessengruppen hinsichtlich einer begrenzten Anzahl von Merkmalskombinationen abgefragt und in tabellarischer Form abgespeichert werden, jedoch ist die maschinelle Weiterverarbeitung in Bezug auf eine bestimmte Forschungsfrage nur möglich, wenn die Informationen, die sich innerhalb der Tabellenfelder befinden, nachträglich und manuell in andere Systeme übertragen werden. Diese Arbeit ist nicht nur umständlich und zeitintensiv, sondern auch sehr fehleranfällig. Die Möglichkeit, Primärdaten für Forschungszwecke verwenden zu können, ist in diesem Konzept nicht vorgesehen.

Allerdings besteht mit der Forschungsdateninfrastruktur BIBB-FDZ seit 2008 eine Möglichkeit zur Mängelminimierung. Hierbei können Mikrodaten aus dem Internet heruntergeladen und in SPSS oder STATA eingebunden werden. Zwar haben wissenschaftlich tätige Personen dabei die Möglichkeit, die vom BIBB erhobenen Mikrodaten für eigene Auswertungen nutzen zu können, allerdings bietet diese Lösung keine Alternative, um eine gleichzeitige synchrone Aktualisierung der Datenbestände in allen Teilsystemen umzusetzen. Sobald sich demnach die Datensätze erweitern – z. B. aufgrund einer Längsschnittuntersuchung –, werden die Anwendungen und Systeme der Dienstnutzer nicht über die Aktualisierung ihrer Datenquelle informiert. Eine stetige Verbindung zwischen dem Analyseinstrument und der Datenquelle über das Internet besteht derzeit noch nicht. Somit können die Veränderungen des Ausbildungsmarktes nicht stetig beobachtet und an die jeweiligen Interessengruppen weiterleitet werden. Des Weiteren bietet das BIBB-FDZ gegenwärtig noch keine Alternative an, um den Austausch von Forschungsdaten zwischen einzelnen wissenschaftlich tätigen Personen zu gewährleisten. Das BIBB allein fungiert als Datenanbieter und bestimmt, welche Mikrodaten bereitgestellt werden.

Daran anknüpfend ergibt sich das Problem, dass unterschiedliche wissenschaftlich tätige Personen und Einrichtungen nicht um die Ergebnisse der anderen wissen. Denn eine Austauschplattform in Form eines beliebig erweiterbaren Datenkataloges existiert nicht. Die Ursache könnte in der Schwierigkeit liegen, systemübergreifende Datenformate zur Verfügung zu stellen, die von unterschiedlichen Systemen mit verschiedenen Eigentümern verarbeitet werden können.

Da sowohl im Rahmen des europäischen Integrationsprozesses als auch im Kontext der internationalen Berufsbildungsforschung die Forderung besteht, Forschungsprogramme miteinander zu vernetzen, um Forschungsergebnisse gemeinsam nutzen und vergleichen zu können, soll diese Forschungsarbeit einen Beitrag zur Mängelminimierung bezüglich der dargestellten Probleme leisten. Hierzu wird das folgende Forschungsziel verfolgt:

- Es soll eine modifizierte, auf Informationstechnologien aufbauende Forschungsdateninfrastruktur entwickelt und erprobt werden, mit der berufs- und ausbildungsbezogene Daten effizienter genutzt werden können als bisher.

- Das konkrete Ziel ist dabei die Entwicklung eines wissenschaftlich fundierten Konzeptes, das einen infrastrukturellen Ansatz aufzeigt, mit dem die Mängel der aktuellen Forschungsdatenstruktur minimiert werden können.

Bis zum jetzigen Zeitpunkt ist noch nicht eindeutig definiert worden, welche Informationstechnologie in diesem Zusammenhang eingesetzt wird. Im Rahmen einer Literaturrecherche, die in Vorbereitung auf diese Forschungsarbeit durchgeführt wurde, hat sich allerdings gezeigt, dass serviceorientierte Architekturen eine akzeptable Grundlage bilden, um die angesprochenen Ziele zu erreichen. Dieser Entscheidung liegen verschiedene Überlegungen zugrunde, die sich in jüngster Zeit angesichts der Vorteile der technologischen Umsetzungsstrategie gezeigt haben. Eine genauere Betrachtung des Sachverhaltes wird im Abschnitt 4.2 beschrieben. Auf der Grundlage der angeführten Überlegungen haben sich die folgenden zwei Forschungsfragen ergeben, die in Bezug auf das hier entwickelte Konzept beantwortet werden sollen:

1. Wie könnte eine serviceorientierte Forschungsdateninfrastruktur im Kontext eines exemplarischen berufswissenschaftlichen Forschungsschwerpunktes gestaltet werden?
2. Wie muss die technologische Infrastruktur beschaffen sein, um die Grundlage für eine plattformunabhängige Verarbeitung und Wiederverwendung von berufswissenschaftlichen Forschungsdaten in unterschiedlichen Analysesystemen zu gewährleisten?

3 Abgrenzung des Forschungsgegenstandes

3.1 Vorbemerkung zur Abgrenzung

Das bereits dargestellte Hauptziel der vorliegenden Untersuchung besteht darin, die Mängel der aktuellen Forschungsdateninfrastruktur BIBB-FDZ durch eine serviceorientierte Forschungsdateninfrastruktur zu minimieren. Es wird daher die Entwicklung eines Konzeptes angestrebt, mit dem die in Abschnitt 3 aufgeworfenen Forschungsfragen beantwortet werden sollen. Ein weiterer Schritt beschäftigt sich anschließend mit der Erprobung des Konzeptes. Genauere Einzelheiten zur methodologischen Grundlage dieser wissenschaftlichen Beweisführung sind dem Abschnitt 5.1 zu entnehmen.

Die Forschungsfragen deuten bereits an, dass das Konzept zur Entwicklung und Einführung einer serviceorientierten Forschungsdateninfrastruktur anhand eines berufswissenschaftlichen

Kontextes erfolgen soll. In Anbetracht der Vielzahl an Forschungsprogrammen macht es Sinn, einen ausgewählten Teilbereich der Berufsbildungsforschung abzugrenzen und diesen im Rahmen der Konzeptentwicklung beizubehalten. Aus diesem Grund bezieht sich die vorliegende Forschungsarbeit im weitesten Sinne auf das Verfahren zur Erstellung und Analyse der Ausbildungsstellenbilanz auf der Basis des Berufsbildungsberichtes. Dieses Verfahren ist ein Bestandteil eines Realweltproblems, das – wie noch zu zeigen sein wird – aus mehreren Unterprozessen besteht. Da sich Realweltprobleme jedoch durch eine hohe Komplexität auszeichnen und die Forschungsarbeit sowohl zeitlichen als auch inhaltlichen Restriktionen unterliegt, muss ein entsprechender Kompromiss getroffen werden. Dieser äußert sich darin, dass eine Reduzierung der Komplexität vorgenommen wird, d. h. das Realweltproblem wird unter modellhaften Annahmen rekonstruiert, sodass ein Rahmen entsteht, auf den sich die weitere Untersuchung stützen kann. (vgl. Hansen / Neumann 2009, S. 286)

Im nachfolgenden Abschnitt geht es zunächst um die Abgrenzung des informationstechnologischen Forschungsgegenstandes und darum, die Begriffe „serviceorientierte Architektur“ und „serviceorientierte Forschungsdateninfrastruktur“ genauer herauszuarbeiten. Anschließend wird Bezug nehmend auf die im Abschnitt 5 aufgestellten Forschungsfragen der berufswissenschaftliche Forschungsgegenstand umrissen.

3.2 Abgrenzung des informationstechnologischen Forschungsgegenstandes

Während der Vorbereitungsphase des Forschungsvorhabens sind verschiedene informationstechnologische Verfahrensweisen diskutiert worden, die zur Realisierung der zuvor definierten Zielstellung eingesetzt werden könnten. Aus diesem Überlegungsprozess heraus entstand zunächst die Notwendigkeit, den Begriff „Informationstechnologien“ abzugrenzen. Rechenberg und Pomberger (vgl. 2002 2002, S. 1040) verstehen unter Informationstechnologien die Gesamtheit aller Arbeits-, Entwicklungs-, Produktions- und Implementierungsverfahren in der Computertechnik einschließlich der zur Informationsproduktion und Kommunikation verfügbaren Techniksysteme. Im vorangegangenen Abschnitt wurde bereits darauf hingewiesen, dass die hier betrachteten Informationstechnologien in engem Zusammenhang mit den Informationssystemen der Berufsbildungsforschung stehen, so z.B. mit der Forschungsdateninfrastruktur, dem Berufsbildungsbericht oder der elektronischen Datenbanken. Aus diesem Grund wird die Definition von Rechenberg und Pomberger weiter eingegrenzt. Unter den hier betrachteten

Informationstechnologien fallen im engeren Sinne analytische, auf Technologie basierende Informationssysteme.

Folgt man den Ausführungen von Chamoni und Gluchowski (2006, S. 10-12), dann wird unter einem analytischen Informationssystem ein unternehmensweites Konzept verstanden, dessen Ziel es ist, eine logisch zentrale, einheitliche und konsistente Datenbasis zur Unterstützung der analytischen Aufgaben von Fach- und Führungskräften zu etablieren. Das Aufbauen eines unternehmensweiten Datenpools, das Benutzen analytischer Werkzeuge zur Entscheidungsunterstützung sowie das Anwenden von Techniken und Verfahren zur Auffindung von bislang verborgenen Mustern und Strukturen in großen Datenmengen sind laut Chamoni und Gluchowski die wesentlichen Bestandteile eines solchen Konzeptes (vgl. ebd., S. 12). Die Autoren verdeutlichen, dass analytische Informationssysteme vor allem im betrieblichen Leistungsprozess verwendet werden, wo sie Aufgaben im Bereich der Kunden-, Lieferanten- und Produktionsfaktorenverwaltung übernehmen und darüber hinaus zur Erfassung, Bearbeitung und Kontrolle von Kundenaufträgen, Lagerbeständen, Produktionsvorgaben beitragen.

In den Veröffentlichungen von Hansen und Neumann (2009, S. 131-133) wird statt eines analytischen Informationssystems der Begriff betriebliches Informationssystem verwendet und definiert. Ihrer Auffassung nach besteht ein betriebliches Informationssystem aus Menschen und Maschinen, die durch Kommunikationstechnologien miteinander in Beziehung stehen. Als betriebliche Systeme werden sie deshalb bezeichnet, weil sie sowohl den Leistungsprozess als auch die Austauschbeziehungen innerhalb eines Betriebes und seiner Umwelt unterstützen. Aus technischer Sicht realisiert ein solches System den Austausch von Zeichenfolgen. Die Autoren sprechen dabei von einer Nachricht. Erst wenn eine Nachricht für einen Empfänger eine bestimmte Bedeutung hat, so die Autoren, wird diese zu einer Information. Sie heben hervor, dass die Information innerhalb des Systems in einer für den Menschen leicht zu verstehenden Form aufbereitet wird, machen aber gleichzeitig auch darauf aufmerksam, dass in der Informationsverarbeitung immer Menschen und Maschinen zusammenwirken. Ein total automatisiertes, gesamtbetriebliches Informationssystem gibt es ihrer Meinung nach nicht.

Bei genauerer Betrachtung ähneln sich beide Definitionen sehr stark. Sowohl bei Chamoni und Gluchowski als auch bei Hansen und Neumann wird auf die Interaktion zwischen Mensch

und Maschine hingewiesen (vgl. Chamoni / Gluchwoski 2006, S. 10-12 und Hansen / Neumann 2009, S. 131-133). Auffällig ist auch der in beiden Definitionen enthaltene Bezug zu betriebswirtschaftlichen Leistungsprozessen. Hierdurch wird deutlich, dass analytische und betriebliche Informationssysteme in erster Linie den Prozess der Gewinnmaximierung bzw. die Effizienzsteigerung unterstützen sollen. Eine solche Prämisse ist im Kontext dieser Arbeit nur insofern von Bedeutung, als durch umfangreiche Informationssysteme eine Optimierung der Berufsbildung und in einem weiteren Schritt die Sicherung des Fachkräftenachwuchses gewährleistet werden kann. Bis zu diesem Punkt ist jedoch eine Vielzahl von Prozessen notwendig, die im Rahmen der bisher vorgestellten Definition keine Berücksichtigung finden. Vielmehr beziehen sich die Definitionen der Autoren auf den Einsatz von analytischen Informationssystemen im Unternehmen. Informationen über Kunden- und Lieferantenbeziehungen, Verbrauchs- und Produktionsfaktoren stellen in vielen Fällen interne Unternehmensgeheimnisse dar, die nicht an die Öffentlichkeit gelangen dürfen. Es liegt deshalb in der Eigenart solcher Systeme, die Offenlegung einer über die Unternehmensgrenzen hinwegreichenden Datenerfassung und Berichterstattung zu vermeiden. In diesem Zusammenhang könnten analytische und betriebliche Informationssysteme nach dem hier dargestellten Verständnis zwar dazu beitragen, die Berufsausbildungsstatistik innerhalb der einzelnen Unternehmen zu erfassen, nicht aber die Berufsbildungsstatistik im Allgemeinen. Um diese Aufgabe zu bewältigen, sind Schnittstellen erforderlich, die einer übergeordneten Instanz den Zugriff auf unternehmensinterne berufsbezogene Daten erlaubt und diese entsprechenden Forschungseinrichtungen wie dem Bundesinstitut für Berufsbildungsforschung oder den Statistischen Landesämtern zur Verfügung stellt. Hieraus ergibt sich folgende Erkenntnis: Analytische bzw. betriebliche Informationssysteme, wie sie hier bisher definiert worden sind, konzentrieren sich auf die Gewinnmaximierung eines einzelnen Unternehmens und sind dadurch für die im Unternehmen anfallenden Aufgaben individuell entwickelt worden. Der Zugriff auf unternehmensinterne Daten ist allenfalls den am Produktionsprozess beteiligten Partnern, wie z. B. Lieferanten und Vertriebskanälen, gestattet. Zudem wirkt sich die relative Geschlossenheit der Systeme nachteilig auf die Schaffung eines gemeinschaftlichen Forschungsraumes aus, weil die Vernetzung mit einem hohen Arbeitsaufwand verbunden ist. In anderen wissenschaftlichen Forschungsbereichen hat sich deshalb eine neue Form eines Informationssystems etabliert. So wird beispielsweise in den Geowissenschaften die so genannte Geodateninfrastruktur eingesetzt. Im Unterschied zur Forschungsdateninfrastruktur des BIBB legt der Gesetzgeber bezüglich der Geodateninfrastruktur explizit fest, dass

innerhalb dieser Struktur die Kombinierbarkeit und die Interaktionsfähigkeit verschiedener Systeme und Techniken unter Einhaltung gemeinsamer Standards gewährleistet werden soll (vgl. § 3 Abs. 4 GeoZG). Unter dem Begriff Geodateninfrastruktur versteht der Gesetzgeber „eine Infrastruktur bestehend aus Geodaten, Metadaten und Geodatendiensten, Netzdiensten und -technologien, Vereinbarungen über gemeinsame Nutzung, über Zugang und Verwendung sowie Koordinierungs- und Überwachungsmechanismen, -prozesse und -verfahren mit dem Ziel, Geodaten verschiedener Herkunft interoperabel verfügbar zu machen“ (§ 3 Abs. 5 GeoZG). Diese Form der Forschungsdateninfrastruktur erscheint für die zuvor betrachtete Problemstellung genau passend zu sein. Dabei weist sie im hohen Maße Ähnlichkeiten zu einer serviceorientierten Architektur auf.

Denn unter einer serviceorientierten Architektur (SOA) wird ein besonderer Ansatz der Systemorganisation verstanden, der darauf abzielt, informationsverarbeitende Dienste so zu entwickeln, dass sie von anderen Systemen und Anwendungen problemlos wiederverwendet werden können (vgl. Kamoun, 2007). Dienste stellen in diesem Zusammenhang grobgranulare Bausteine von Softwaresystemen dar, die sich flexibel in komplexe Geschäftsprozesse einflechten und auch wieder lösen lassen. Demnach sollen Institutionen und Organisationen Dienste unterschiedlichster Anbieter aus einem bestehenden Verzeichnis auswählen und beliebig innerhalb der eigenen Anwendung miteinander kombinieren können. Dabei steht die Integration von Diensten in implementierungsunabhängiger Form im Vordergrund, d. h. Dienste können unabhängig von einer bestimmten Programmiersprache oder Anwendungssoftware verwendet werden. Durch die Kombination zweier Dienste entsteht wiederum ein neuer Dienst, der ebenfalls über ein bestehendes Verzeichnis angeboten werden kann (vgl. Schill / Springer 2012, S. 20-21). Im weiteren Verlauf der Arbeit sollen die Begriffe Service, Webservice und Dienste synonym verwendet werden.

3.3 Abgrenzung des berufspädagogischen Forschungsgegenstandes

In den vorangegangenen Ausführungen wurde bereits darauf eingegangen, dass sich der Einsatz des informationstechnologischen Forschungsgegenstandes auf die Umsetzung einer serviceorientierten Forschungsdateninfrastruktur im Kontext eines berufswissenschaftlichen Forschungsschwerpunktes bezieht. Anknüpfend an die erste Forschungsfrage erfolgt nun eine genauere Abgrenzung des berufswissenschaftlichen Forschungsschwerpunktes. Wie bereits

erwähnt, wird hier das Ziel verfolgt, einen Teil der Berufsbildungsforschung unter modellhaften Annahmen darzustellen. Der Verein Deutscher Ingenieure versteht unter einem Modell „eine vereinfachte Nachbildung eines geplanten oder existierenden Systems mit seinen Prozessen in einem anderen begrifflichen oder gegenständlichen System“ (2000, S. 3). „Es unterscheidet sich hinsichtlich der untersuchungsrelevanten Eigenschaften nur innerhalb eines vom Untersuchungsziel abhängigen Toleranzrahmens vom Vorbild“ (ebd.). Den Ausgangspunkt bildet die Ausbildungsstellenbilanz, die Teil der Berufsbildungsberichte ist. Ein erster Schritt zur Erstellung der Ausbildungsstellenbilanz ist die Zählung und Analyse der Lehrstellen sowie der neu abgeschlossenen Ausbildungsverträge (vgl. Krekel 2006, S. 208). Diese Prozesse bilden im weiteren Verlauf der Arbeit den exemplarischen berufswissenschaftlichen Forschungsschwerpunkt, an dem der Einsatz einer serviceorientierten Forschungsdateninfrastruktur erprobt werden soll. Die in diesem Zusammenhang benötigten Daten beziehen sich deshalb auf die Erhebungsmerkmale der Berufsbildungsstatistik gemäß § 88 Abs. 1 Nr. 1-5 BBiG. Eine Übersicht über die Erhebungsmerkmale gibt die Abbildung 2.

§ 88 Erhebungen

(1) Die jährliche Bundesstatistik erfasst

1.
für jeden Auszubildenden und jede Auszubildende:
 - a) Geschlecht, Geburtsjahr, Staatsangehörigkeit;
 - b) allgemeinbildender Schulabschluss, vorausgegangene Teilnahme an berufsvorbereitender Qualifizierung oder beruflicher Grundbildung, berufliche Vorbildung;
 - c) Ausbildungsberuf einschließlich Fachrichtung;
 - d) Ort der Ausbildungsstätte, Wirtschaftszweig, Zugehörigkeit zum öffentlichen Dienst;
 - e) Ausbildungsjahr, Abkürzung der Ausbildungsdauer, Dauer der Probezeit;
 - f) Monat und Jahr des Beginns der Berufsausbildung, Monat und Jahr der vorzeitigen Auflösung des Berufsausbildungsverhältnisses;
 - g) Anschlussvertrag bei Stufenausbildung mit Angabe des Ausbildungsberufs;
 - h) Art der Förderung bei überwiegend öffentlich, insbesondere auf Grund des Dritten Buches Sozialgesetzbuch geförderten Berufsausbildungsverhältnissen;
 - i) Monat und Jahr der Abschlussprüfung, Art der Zulassung zur Prüfung, Monat und Jahr der Wiederholungsprüfung, Prüfungserfolg;
2.
für jeden Prüfungsteilnehmer und jede Prüfungsteilnehmerin in der beruflichen Bildung mit Ausnahme der durch Nummer 1 erfassten Auszubildenden:
Geschlecht, Geburtsjahr, Berufsrichtung, Vorbildung, Wiederholungsprüfung, Art der Prüfung, Prüfungserfolg;
3.
für jeden Ausbilder und jede Ausbilderin:
Geschlecht, Geburtsjahr, Art der fachlichen Eignung;
4.
für jeden Ausbildungsberater und jede Ausbildungsberaterin:
Geschlecht, Geburtsjahr, Vorbildung, Art der Beratertätigkeit, fachliche Zuständigkeit, durchgeführte Besuche von Ausbildungsstätten;
5.
für jeden Teilnehmer und jede Teilnehmerin an einer Berufsausbildungsvorbereitung, soweit der Anbieter der Anzeigepflicht des § 70 Abs. 2 unterliegt:
Geschlecht, Geburtsjahr, Staatsangehörigkeit, Berufsrichtung.

Abbildung 2: Erhebungsmerkmale im Kontext der Berufsbildungsstatistik. Entnommen aus dem Berufsbildungsgesetz.

Der Entstehungsprozess der Ausbildungsstellenbilanz wird von einer Vielzahl von Faktoren beeinflusst, die im Rahmen dieser Arbeit nicht alle berücksichtigt werden können. Aus diesem Grund wird das Realweltproblem weiter eingegrenzt:

- Es wird nur das duale System der Berufsbildung betrachtet.
- Die zu erfassenden Merkmale der Auszubildenden basieren auf § 88 Abs. 1 Nr. 1. a), b), c) und f) BBiG. Allerdings ist für § 88 Abs. 1 Nr. 1. b) BBiG eine engere Eingrenzung notwendig und zwar dahingehend, dass in diesem Modell nur die berufliche Vorbildung in Form einer dualen Berufsausbildung von Bedeutung ist. Jede andere Form der beruflichen Vorbildung ist unzulässig. Maßgeblich ist ein in der Vergangenheit liegendes Ausbildungsverhältnis bzw. ein in der Vergangenheit liegender Vertragsschluss. Es wird nicht berücksichtigt, ob das Ausbildungsverhältnis erfolgreich oder erfolglos beendet wurde. Eine weitere Eingrenzung wird im § 88 Abs. 1 Nr. 1 f) BBiG vorgenommen, indem nur das Jahr des Beginns bzw. des Abbruchs der Berufsausbildung erfasst wird. Der Monat bleibt unberücksichtigt. Ebenfalls unberücksichtigt bleiben alle Personenmerkmale, die hier nicht explizit aufgeführt wurden.
- Die zu erfassenden Merkmale der Ausbildungsstätten basieren auf § 88 Abs. 1 Nr. 1 d) BBiG, wobei das Merkmal „Zugehörigkeit zum öffentlichen Dienst“ keine Berücksichtigung findet. Stattdessen wird das Merkmal „Name der Ausbildungsstätte“ eingeführt.
- Es wird die Idealbedingung angenommen, dass alle Ausbildungsplatznachfrager/-innen innerhalb einer betrachteten Periode eine Ausbildungsstelle erhalten.
- Ausbildungsnachfrager/-innen setzen sich zusammen aus Personen, die zum ersten Mal eine Berufsausbildung beginnen, und Personen, die bereits eine Ausbildung im dualen System absolviert bzw. begonnen haben.

Um die erste Forschungsfrage beantworten zu können, soll ein Konzept entwickelt werden, welches das Verfahren zur Lehrstellenzählung sowie zur Ermittlung der neu abgeschlossenen Ausbildungsverträge unter Einbeziehung einer bestimmten Dienstkombi-
Anhand dieses Konzeptes soll geprüft werden, ob sich der Einsatz einer serviceorientierten Forschungsdateninfrastruktur besser zur Bereitstellung von forschungsrelevanten Daten eignet als die gegenwärtigen Informationssysteme der Berufsbildungsforschung. Dabei sind die im

Abschnitt 3 beschriebenen Mängel der gegenwärtigen Informationssysteme aus dem Bereich der Berufsbildungsforschung zu berücksichtigen.

Des Weiteren soll auch die zweite Forschungsfrage hinsichtlich der eingangs formulierten Zielstellung abgegrenzt werden. Das Konzept einer serviceorientierten Berufsbildungsforschung wird hier verstanden als die Fähigkeit, mittels Informationstechnologien einen gemeinschaftlichen Forschungsraum abzubilden, der verschiedenen heterogenen Systemen eine Schnittstelle bietet, damit einzelne wissenschaftlich tätige Personen an den qualitativen Forschungsergebnissen der anderen partizipieren können. Um dies zu realisieren, sind standardisierte Datenformate und Datentypen erforderlich, die von den heterogenen Systemen interpretiert werden können. Zur Beantwortung der Forschungsfrage wird die Entwicklung von standardisierten Datentypen angestrebt. Dabei beziehen sich die Datentypen auf die zuvor definierten Erhebungsmerkmale.

4 Methoden

4.1 Vorbemerkung zur Methodik

Um die Forschungsfragen beantworten zu können, wird eine Machbarkeitsstudie erarbeitet, die den Umstellungsprozess von der gegenwärtigen Berufsbildungsforschung zur serviceorientierten Berufsbildungsforschung unter Idealbedingungen, d. h. unter modellhaften komplexitätsreduzierenden Annahmen, beschreibt. Die methodische Umsetzung soll dabei in zwei Schritten durchgeführt werden: Im ersten Schritt wird ein Migrationskonzept erarbeitet. Dabei bedarf es wiederum eines methodischen Bezuges, der zwei Aspekte vereint: den ingenieurwissenschaftlichen und den berufswissenschaftlichen. Der ingenieurwissenschaftliche Aspekt bezieht sich auf Fragen bezüglich des Einsatzes von Methoden und Verfahren zur Systementwicklung, Standardisierung, Dienstkomposition und Systemmigration. Während der Vorbereitungsphase hat sich allerdings gezeigt, dass hierzu zwei verschiedene wissenschaftliche Bezugsquellen in Erwägung gezogen werden können: Auf der einen Seite besteht die Möglichkeit, Methoden und Verfahren aus der Fachwissenschaft der Systemtechnik anzuwenden, auf der anderen Seite existieren eine Vielzahl von empirischen Studien, in denen der Umstellungsprozess zu einer serviceorientierten Architektur praxisnah untersucht worden ist. Welche der beiden Bezugsquellen in der vorliegenden Arbeit verwendet wurde, wird im Abschnitt 5.2.1 näher begründet. Der berufswissenschaftliche

Aspekt wird dagegen benötigt, um die Aufgaben der Dienste aus einer fachlich korrekten Sichtweise zu beschreiben. Hierbei wurde eine rein hermeneutische Vorgehensweise gewählt. Die Grundlage bildet dabei wissenschaftliche Fachliteratur zur Methodik der Berufsbildungsforschung.

In einem zweiten Schritt der methodischen Umsetzung soll das zuvor erarbeitete Migrationskonzept unter Einhaltung der modellhaften Annahmen aus Abschnitt 4 simuliert werden. Ziel der Simulation ist es, Erkenntnisse darüber zu gewinnen, ob die hier erarbeitete serviceorientierte Berufsbildungsforschung tatsächlich die zuvor definierten Ziele erfüllt.

Angesichts einer solchen interdisziplinären Forschung ist die Frage nach der Legitimität eines solchen Ansatzes nahezu berechtigt. Dass aus der Sicht der Berufsbildungsforschung dieser wissenschaftliche Spagat explizit gefordert wird, belegen die Ergebnisse des Forschungsprogramms „Methodologie der Berufsbildungsforschung“, das 1971 noch vom damaligen Bundesinstitut der Berufsbildungsforschung (BBF) durchgeführt worden war. Dort kam man zu der Erkenntnis, dass die Entwicklung eines methodologischen und terminologischen Instrumentariums für die Arbeiten des Institutes unter Berücksichtigung der interdisziplinären Zusammenhänge erfolgen muss (vgl. Rauner 2006, S. 10). Bis heute besteht Einigkeit darüber, dass Berufsbildungsforschung nur interdisziplinär organisiert werden kann: „vorrangig durch die Beteiligung von Fächern wie Psychologie, Industrie-, Arbeits-, Bildungs- und Betriebssoziologie, Natur- und Ingenieurwissenschaften, Erziehungs- und Wirtschaftswissenschaften“ (Rauner 2006, S. 10). Dabei bedienen sich Wissenschaftler „je nach ihrer disziplinären Herkunft im Repertoire der disziplinspezifischen Methoden und konstituieren damit gleichermaßen auch ihren Gegenstand als einen sozial-, arbeits-, ingenieur- oder naturwissenschaftlichen“ (ebd., S. 14).

4.2 Migrationskonzept

4.2.1 Vorgehensweise

Um die Forschungsfragen beantworten zu können, ist die Entwicklung und Erprobung eines Migrationskonzeptes geplant, welches den Umstellungsprozess von der gegenwärtigen Berufsbildungsforschung zu einer serviceorientierten Berufsbildungsforschung beschreibt. Unter Migration wird die Umstellung bzw. der Übergang von einem System auf das andere bezeichnet (vgl. Stahlknecht / Hasenkamp 2005, S. 75). Weil sich jedoch die Entwicklung von

komplexen Informationssystemen nicht schon zu Projektbeginn als Ganzes planen lässt, sind eine Vielzahl von Methoden und Verfahren entwickelt worden, die in der ingenieurwissenschaftlichen Disziplin der Systemtechnik verortet sind (vgl. Stahlknecht / Hasenkamp 2005, S. 215; Brügge / Dutoit 2004, S. 25). Insbesondere der Bereich der Softwareentwicklung wird innerhalb der Fachwissenschaft umfangreich erforscht. Aus diesem Grund entstand die Überlegung, das Migrationskonzept auf der Grundlage eines Softwareentwicklungsprozesses zu erarbeiten. Um jedoch keinen voreiligen Entschluss zu treffen, sollen zunächst die Methoden und Verfahren der Softwareentwicklung vorgestellt und anschließend ihre Eignung bezüglich der hier verfolgten Fragestellung geprüft werden.

Die Termini „Softwareentwicklung“ bzw. „System Engineering“ werden im Folgenden genauer definiert, sodass für die geplante Analyse ein einheitliches Begriffsverständnis gegeben ist. Balzert (2001) versteht unter dem Begriff „Softwareentwicklung“ die „zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden, Konzepten, Notationen und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Softwaresystemen“ (ebd., S. 36). An dieser Stelle sei darauf hingewiesen, dass der Terminus „Systementwicklung“ nicht gleichgesetzt wird mit „Systementwurf“. Weil die Systementwicklung eine Reihe von Entwicklungsaktivitäten umfasst, entspricht sie vielmehr einer Strategie für die Durchführung eines Projektes (vgl. Zuser et al. 2004, S. 70; Brügge / Dutoit 2004, S. 40). Im Gegensatz dazu ist der Systementwurf ein Bestandteil dieser Strategie (vgl. Brügge / Dutoit 2004, S. 272), bei dem jene Teilaspekte des Systems definiert werden, die für eine unmissverständliche Implementierung notwendig sind (vgl. Zuser et al. 2004, S. 272). In diesem Abschnitt wird ebenfalls von einer Strategie ausgegangen, wenn von der Software- bzw. Systementwicklung gesprochen wird.

Stahlknecht und Hasenkamp verknüpfen die Systementwicklung mit dem Konzept des System Engineering, welches 1950 von den Bell Laboratorien erarbeitet wurde (vgl. Stahlknecht und Hasenkamp 2005, S. 208-222). Demnach orientiert sich der Entwicklungsprozess an mehreren zeitlich aufeinander folgenden Phasen. Systemanalyse, Systementwicklung, Systemeinführung und Systempflege bilden laut Stahlknecht und Hasenkamp die vier Grundphasen des Konzeptes. Die Autoren betonen jedoch, dass die strenge Einhaltung des Phasenkonzeptes nicht immer möglich ist, weshalb ihrer Meinung nach allgemein von einem Vorgehensmodell gesprochen werden kann. Hierunter verstehen

sie „jede modellhafte, abstrahierende Beschreibung von Vorgehensweisen, Richtlinien, Empfehlungen oder Prozessen, die für einen abgegrenzten Problembereich gelten und in einer möglichst großen Anzahl von Einzelfällen anwendbar sind“ (ebd. S. 215). Fast alle heutigen Vorgehensmodelle, so die Autoren, orientieren sich an den vier Phasen Analyse, Entwurf, Realisierung und Einführung.

Hansen und Neumann berufen sich implizit ebenfalls auf Vorgehensmodelle (vgl. 2009, S. 364-372). Allerdings differenzieren sie den Begriff hinsichtlich zweier Merkmale: So lässt sich ihrer Auffassung zufolge die Vorgehensweise beim Entwicklungsprozess entweder nach der Art der Phasendurchläufe oder nach dem Projektumfang unterscheiden. Wird der Projektumfang als Entscheidungsgrundlage in Betracht gezogen, hängt die einzusetzende Methode von der Teamgröße ab. Die agile Entwicklungsmethode, das V-Modell oder Scrum können laut Hansen und Neumann dabei zum Einsatz kommen; da im Rahmen dieser Forschungsarbeit jedoch kein Team existiert, soll hier eine Abgrenzung hinsichtlich des Untersuchungsumfangs vorgenommen werden.

Um die Betrachtung der Methodik nicht unnötig auszudehnen, wendet sich die weitere Untersuchung den Vorgehensmodellen zu, die sich nach der Art der Phasendurchläufe unterscheiden lassen. Anders verhält es sich, wenn die Art der Phasendurchläufe als Entscheidungsgrundlage dient. Folgt man den Ausführungen von Hansen und Neumann, kann sich in diesem Fall die Vorgehensweise an einer sequenziellen oder iterativen Methode orientieren. Nach der sequenziellen Methode werden die Phasen nacheinander durchgeführt. Erst wenn die zeitlich vorangegangene Phase abgeschlossen ist, kann eine neue Phase begonnen werden. Jede Phase wird demzufolge genau einmal durchlaufen. Ein Vertreter des sequenziellen Phasenmodells ist das klassische Wasserfallmodell. Dies sieht nach Meinung der Autoren eine Abfolge der Arbeitsschritte Anforderungsanalyse, Entwurf, Implementierung, Test und Wartung vor. Diesem Vorgehen liegt laut Hansen und Neumann die Annahme zugrunde, dass „die Software primär ‚neu‘ entwickelt wird und dass die Software nach dem Abschluss des Entwicklungsprozesses ‚fertig‘ ist“ (ebd., S. 369), wodurch alle Anforderungen zu Beginn der Entwicklungsphase vollständig erfasst und beschrieben werden müssen. Zuser, Grechenig und Köhle weisen darauf hin, dass sich das Wasserfallmodell gut für Projekte eignet, in denen die Arbeitsschritte deutlich voneinander getrennt werden können (vgl. 2004, S. 71). Darüber hinaus stellen sie fest, dass insbesondere Projekte mit kleinen Arbeitsgruppen nach dem Wasserfallmodell vorgehen sollten, da alle Mitarbeiter gleichzeitig

an einen Arbeitsschritt arbeiten können. Anderer Auffassung sind dagegen Hansen und Neumann (vgl. 2009, S. 367). Sie argumentieren, dass Wasserfallmodelle in der Praxis unrealistisch sind, weil für die Benutzer greifbare Ergebnisse erst zu einem relativ späten Zeitpunkt vorliegen, wodurch es sowohl aus technischen als auch aus wirtschaftlichen Gründen kaum mehr möglich ist, Änderungswünsche der künftigen Benutzer zu berücksichtigen. Die Autoren weisen implizit auf die Gefahr hin, dass fertige Systeme bereits vor ihrer Einführung obsolet werden könnten. Dem entgegen steht laut Hansen und Neumann die inkrementelle Methode (vgl. ebd., S. 369). Obwohl sie ebenfalls zu den sequenziellen Phasenmodellen gehört, unterscheidet sie sich vom Wasserfallmodell hinsichtlich der paradigmatischen Entwurfsvorstellung: Anstatt zu einem gewissen Zeitpunkt ein komplettes Softwareprodukt auszuliefern, ohne Zwischenergebnisse mit den Benutzern abzugleichen, werden bei der inkrementellen Methode in regelmäßigen Abständen sog. Prototypen entwickelt, wodurch einzelne Teilsysteme entstehen, die schrittweise vervollständigt und schlussendlich zu einem Gesamtsystem integriert werden. Zuser, Grechenig und Köhle behaupten, dass es mit diesem Modell möglich ist, besser und schneller auf Anforderungsänderungen reagieren zu können, weil das Produkt in vielen kleinen überschaubaren Schritten erzeugt wird (vgl. 2004, S. 75). Allerdings weisen die Autoren auf das Risiko hin, hinzukommende Anforderungen nicht mehr in die gegebene Architektur integrieren zu können, weil sich die Systementwicklung in einem fortgeschrittenen Stadium befindet. Sowohl Hansen und Neumann als auch Stahlknecht und Hasenkamp unterstreichen bei rein inkrementellen Methoden die Nachteile, die mit sequenziellen Phasenmodellen einhergehen: Mögliche Anforderungen, die während des Entwicklungsprozesses auftreten und sich beispielsweise durch Marktveränderungen oder anderen Umweltfaktoren ergeben, können beim sequenziellen Vorgehen nicht berücksichtigt werden (vgl. Hansen / Neumann 2009, S. 370; Stahlknecht / Hasenkamp 2005, S. 218-219). Eine Möglichkeit zur Mängelminimierung sehen die Autoren in der iterativen Methode. Hansen und Neumann erklären, dass dabei die einzelnen Phasen der Entwicklung mehrfach durchlaufen werden, wodurch die Nachteile sequenzieller Phasenmodelle kompensiert würden (vgl. 2009, S. 370-371). Ihrer Meinung nach kann das Prinzip der iterativen Methode anhand des Spiral-Modells verdeutlicht werden. Der Interpretationsversuch von Brügge und Dutoit (2004, S. 655) stützt sich auf die Ausführungen von Boehm (1987, S. 128-142), der als Entwickler des Spiral-Modells in die Fachliteratur einging. Ihren Ausführungen zufolge vollzieht sich der Entwicklungsprozess zyklisch, wobei jeder Zyklus aus vier Phasen besteht. Während die Entwickler in der ersten Phase Anforderungen, Alternativen, Randbedingungen und Ziele des

Systems definieren, wird laut Brügge und Dutoit in der zweiten Phase eine Risikoanalyse durchgeführt. Darauf aufbauend wird in der dritten Phase den Autoren zufolge die Entwicklung eines Prototyps angestrebt, der mit den in dieser Iteration erarbeiteten Ergebnissen in Beziehung steht. Die letzte Phase widmet sich schließlich der Planung des nächsten Zyklus. Eine besondere Stellung nehmen laut Hansen und Neumann die Prototypen ein (vgl. 2009, S. 371). Diese passen sich, so die Autoren, Schritt für Schritt sowohl den Idealvorstellungen des Benutzers als auch veränderten Umweltfaktoren an (vgl. ebd.). Zuser Grechenig und Köhle erklären, dass in jedem Zyklus genau ein Prototyp gebaut wird, der dann eine realitätsnahe Beurteilung des zu entwickelnden Systems erlaubt. Ihrer Meinung nach stellen die dabei entstehenden Prototypen – im Unterschied zur inkrementellen Methode – noch keine funktionsfähige Softwarekomponente dar (vgl. 2004, S. 73-75). Die Autoren sind vielmehr der Meinung, dass sich iterative Phasenmodelle, zu denen auch das Spiral-Modell gehört, aufgrund der umfangreichen Risikoanalysen eher für komplexe Projekte eignen (vgl. ebd., S. 73). Außerdem machen sie deutlich, dass sich die Anzahl der Durchläufe erst während des Projektes ergibt und durch auftretende Risiken beeinflusst wird, wodurch zu Beginn des Projektes ein Zeit- und Kostenplan nur schwer zu erstellen ist. Auch die Risikoanalyse selbst ist nur von erfahrenen Projektleitern durchführbar, weil eine Fehlanalyse das Projekt unnötig verlängern kann (vgl. ebd.).

Die Frage, ob sich die hier dargestellten Vorgehensmodelle als Strategie für die Überführung der Berufsbildungsforschung in eine serviceorientierte Berufsbildungsforschung eignen, beeinflusst das weitere methodische Vorgehen. Um diese Frage beantworten zu können, sollen die o. g. Modelle hinsichtlich der hier verfolgten Zielstellung untersucht werden.

Die sequentielle Vorgehensweise mit dem Wasserfallmodell als Referenzmodell wirkt auf den ersten Blick verständlich und verleiht dem methodischen Vorgehen eine überschaubare Struktur. Wie bereits erwähnt, kann diese Forschungsarbeit nicht dem Anspruch an eine Gruppenleistung gerecht werden. Aber wenngleich die methodische Umsetzung in Form einer Einzelleistung erfolgt, existiert mit dem Wasserfallmodell im Rahmen dieser modellhaften Untersuchung ein akzeptabler Ansatz. Laut Zuser, Grechenig und Köhle eignet sich dieses Modell besonders für kleine Arbeitsgruppen (vgl. 2004, S. 71), was seine Bedeutung unterstreicht und wodurch es dem Anliegen der vorliegenden Untersuchung am ehesten gerecht wird. Der Nachteil dieser Methode ist allerdings, dass bei einer realen Umsetzung keine funktionsfähigen Zwischenprodukte getestet werden. Ob die SOA alle Anforderungen

erfüllt, würde sich demzufolge erst in der Phase der Realisierung zeigen, was zur Folge hätte, dass der gesamte Entwurf inklusive aller Dienste, Standards und Schnittstellen geändert werden müsste. Des Weiteren setzt eine reale Umsetzung nach dem Wasserfallmodell voraus, dass schon zu Beginn des Projektes alle Systeme, Dienste, Anwendungen und Geschäftsprozesse eindeutig identifiziert worden sind und keinen äußerlichen Veränderungsprozessen unterliegen, da ansonsten die Behauptung von Hansen und Neumann greift: Das System wäre obsolet, noch bevor es realisiert worden ist (vgl. 2009, S. 367). Weil man jedoch mit einer SOA einen Ansatz verfolgt, dessen Vorteile insbesondere in einer Verbesserung der Flexibilität liegen, widerspricht das Wasserfallmodell in seiner Standardform dem Prinzip einer SOA. Schlussfolgernd kann daher festgehalten werden, dass sich das Wasserfallmodell gut für die hier verfolgte methodische Begleitung eignet. Es ist überschaubar und überschreitet nicht den methodischen Rahmen. Bezüglich einer realen Migration sprechen viele Argumente gegen das Wasserfallmodell. Da es sogar dem SOA-Prinzip entgegenwirkt, ist es als Strategie zur SOA-Implementation ungeeignet.

Eine Alternative stellt die inkrementelle Methode dar. Obwohl die Entwicklungsphasen dabei nicht wiederholt durchlaufen werden, wird der Entwicklungsprozess so gestaltet, dass in regelmäßigen Abständen eine funktionsfähige Systemkomponente in Form eines Prototyps für die Benutzung zur Verfügung steht. Hansen und Neumann behaupten allerdings, dass die inkrementelle Methode ebenso wie das Wasserfallmodell nicht in der Lage sei, veränderte Anforderungen und Umweltfaktoren während des Entwicklungsprozesses zu berücksichtigen. Gegen diese Ansicht lassen sich folgende Argumente anführen: Zum einen kann jederzeit eine neue Systemfunktion in Form eines Prototyps implementiert werden, zum anderen durchläuft jeder Prototyp während seiner Entwicklung die Phasen Analyse, Entwurf und Implementierung, wodurch der Unterschied zum Wasserfallmodell deutlich wird. Während das klassische Wasserfallmodell genau einmal durchlaufen wird, besteht die inkrementelle Methode aus mehreren Durchläufen, in denen wiederum die Phasen des Wasserfallmodells abgearbeitet werden können. Folglich kann festgehalten werden, dass sich für die reale Umsetzung einer SOA durchaus die inkrementelle Entwicklungsmethode eignet, weil sie laut Hansen und Neumann die schrittweise Weiterentwicklung von Prototypen unterstützt. So können jederzeit neue Dienste inkrementell hinzugefügt werden, ohne dass zu Beginn des Projektes alle Anforderungen an das System erfasst werden müssen. Hierdurch reduziert sich gleichzeitig auch das Risiko einer Kostenexplosion, weil weniger Investitionen in die Middleware und die Anforderungsanalyse getätigt werden müssen. In Bezug auf das hier

verfolgte modellhafte Migrationskonzept bleibt allerdings zu fragen, wie eine derartige Strategie dokumentiert und beschrieben werden kann, ohne dass der Umfang dieser Arbeit überschritten wird. Jedem einzelnen Dienst eine eigene Migration zu widmen, die darüber hinaus entsprechende Systementwürfe in Form einer UML enthält, ist eine anspruchsvolle Aufgabe, die sehr viel Zeit in Anspruch nimmt. Aus diesem Grund ist die inkrementelle Methode für eine reale Umsetzung zwar geeignet, musste aber im Rahmen dieser Forschungsarbeit verworfen werden.

Die iterative Methode berücksichtigt entgegen den bereits vorgestellten Modellen das wiederholte Durchlaufen einzelner Phasen. Im Unterschied zur inkrementellen Methode wird derselbe Prototyp evolutionär weiterentwickelt und verbessert, ohne dass nach jedem Zyklus eine neue Systemfunktion in Form einer eigenständigen Komponente bereitgestellt wird. Bei dieser Methode wird besonders der enge Kontakt zwischen Entwicklern, Benutzern und Auftraggebern deutlich. Im Rahmen dieser Arbeit existieren derartige Interessengruppen allerdings nicht, wodurch eine iterative Weiterentwicklung ausscheidet. Anders verhält es sich bei der realen Umsetzung einer SOA. Besonders bei der Migration einer Forschungsdateninfrastruktur müssen Dienstentwickler eng mit wissenschaftlich tätigen Personen zusammenarbeiten.

Die bisher vorgestellten Strategien aus dem Bereich der Softwareentwicklung haben gemeinsam, dass jeweils der Lebenszyklus eines konkreten Softwareproduktes betrachtet wird. Da eine SOA aus mehreren autark arbeitenden Softwarekomponenten besteht, die nach dem Baukastenprinzip organisiert sind, besteht die Gefahr, dass ein Migrationskonzept auf der Basis o. g. Vorgehensmodelle unüberschaubar wird; jeder einzelne Dienst müsste hier nach einem der drei Vorgehensmodelle beschrieben werden. Anzahl, Umfang und Funktionalität der Dienste sind zum jetzigen Zeitpunkt jedoch noch nicht bekannt. Darüber hinaus sollte das Migrationskonzept aus einer abstrakten Sichtweise betrachtet werden und sich nicht schwerpunktmäßig auf die Implementierung ausgewählter Dienste beziehen; es sollte also vielmehr einen allgemeingültigen Machbarkeitsnachweis erbringen.

Eine andere Möglichkeit besteht deshalb darin, empirische Studien als Bezugsquelle in Betracht zu ziehen, in denen die Entwicklung und Migration serviceorientierter Architekturen in der Praxis untersucht worden sind. Eine ausführliche qualitative Studie wurde kürzlich von

Nasr, Gross und Van Deursen (2011, Übers. D. Verf.)¹ veröffentlicht. Die Autoren verglichen dabei das Migrationskonzept zweier Unternehmen und stellten darauf aufbauend einen „Best practice“-Ansatz auf. Dieser ist so formuliert, dass die konkrete Ausgestaltung der Dienste abstrakt gehalten wird, d. h. er stellt eine Sammlung von Erfahrungswerten dar. Im Folgenden sollen die wichtigsten Punkte des Ansatzes vorgestellt werden.

1. Zunächst belegen die Untersuchungen der Autoren, dass die inkrementelle Entwicklungsmethode eine empfehlenswerte Strategie zur Migration darstellt. Dabei können ein oder mehrere Dienste den Ausgangspunkt bilden. Beginnt die Migration schrittweise mit nur einem Dienst bzw. einem System, so haben alle Interessengruppen den Untersuchungen zufolge die Möglichkeit, den Migrationsprozess besser nachvollziehen zu können.
2. Dem Identifizieren der Dienste sollte laut Untersuchungen ausreichend Zeit gewidmet werden. Empfehlenswert ist dabei eine zentrale Verwaltungsstelle, mit der Dienste systematisch geordnet werden können. Das Erstellen eines Service-Portfolios hat sich in diesem Zusammenhang bewährt.
3. Es zeigte sich, dass sich die Visualisierung einer SOA anhand eines Web-Portals vornehmen lässt, sodass auch Personen ohne technisches Hintergrundwissen das Funktionsprinzip einer SOA verstehen.
4. Die Untersuchungen belegen zudem, dass das verfügbare Budget nicht schon in der Anfangsphase für die komplette Infrastruktur ausgegeben werden sollte, da es sich erwiesen hat, dass einige Dienste, die zu Beginn geplant werden, im Laufe des Migrationsprozesses an Bedeutung verlieren. Stattdessen empfiehlt es sich, in kleinen Schritten die SOA zu erweitern.

Bei der Gegenüberstellung beider Bezugsquellen wird deutlich, dass hier zwei verschiedene Betrachtungsweisen zugrunde liegen. Während Softwareentwicklungsprozesse eine Strategie darstellen, offenbaren die Studien von Nasr, Gross und Van Deursen (2011, Übers. D. Verf.) praxisnahe Empfehlungen, mit denen Migrationsprozesse erfolgreich in der Praxis umgesetzt werden können. Würde man sich im Rahmen dieser Untersuchung ausschließlich auf Softwareentwicklungsprozesse beziehen, wäre demzufolge nicht sichergestellt, dass das daraus resultierende Migrationskonzept auch tatsächlich valide ist. Es würde dem Konzept an einer wissenschaftlichen Beweisführung fehlen. Erst durch die Kombination beider Bezugsquellen erhält man ein Konzept, das einerseits einer klaren Strukturierung folgt und

¹ Im Originaldokument sind keine Seitenzahlen angegeben. Die Untersuchungen stehen als Anlage im PDF-Format auf der beigelegten CD-ROM zur Verfügung.

andererseits sicherstellt, dass die hier aufgezeigte Strategie wissenschaftlich fundiert ist. In den vorangegangenen Ausführungen wurde bereits dargelegt, dass mit dem Wasserfallmodell eine klare Struktur verfolgt werden kann, die im Rahmen dieser Arbeit realisierbar ist. In dem geplanten Migrationskonzept wird deshalb die Struktur des Wasserfallmodells beibehalten – allerdings fließen in den einzelnen Phasen die Untersuchungsergebnisse von Nasr, Gross und Van Deursen (2011, Übers. D. Verf.) ein. Dabei wird wie folgt vorgegangen:

In der Analysephase wird eine Geschäftsprozessmodellierung durchgeführt. Diese bezieht sich auf den Dienstanbieter. Von diesem ist nur bekannt, dass er das hier entwickelte Service-Portfolio bereitstellt. Daran schließt sich der Entwurf an. Im Entwurf wird nicht der Dienstanbieter oder der Dienstanwender betrachtet; vielmehr wird die gesamte Architektur der serviceorientierten Forschungsdateninfrastruktur analysiert. Mittels eines Web-Portals soll dabei zum einen eine zentrale Verwaltungseinheit eingerichtet werden, die ein Service-Portfolio enthält, das in Bezug auf die hier verfolgte berufswissenschaftliche Problemstellung angemessen ist, zum anderen soll sie dazu dienen, die Funktionsweise der serviceorientierten Dateninfrastruktur für Interessengruppen durchlässiger darzustellen. In der Realisierungsphase werden dieselben Technologien eingesetzt, die sich auch in den Untersuchungen von Nasr, Gross und Van Deursen (2011, Übers. D. Verf.) bewährt haben. Die Entwicklung und Spezifikation der Services und Datentypen erfolgt wieder aus der Sichtweise des Dienstansbieters.

4.2.2 Analyse

Der Aufbau einer serviceorientierten Architektur wird regelmäßig mit der Analyse von Geschäftsprozessen in Verbindung gebracht. Obwohl Geschäftsprozessanalysen und serviceorientierte Architekturen aus zwei völlig verschiedenen Kontexten heraus entstanden sind, werden beide Techniken oft miteinander kombiniert. Während die Geschäftsprozessanalyse eine Managementaufgabe ist, die dazu dient, unternehmensübergreifende Aktivitäten zwischen Lieferanten und Kunden zu identifizieren, bilden serviceorientierte Architekturen technologiebasierende Ansätze um unterschiedliche, autark arbeitende Softwarekomponenten miteinander zu verbinden, sodass sie von verschiedenen Anwendungen wiederverwendet werden können. Werden beide Techniken miteinander kombiniert, so entsteht ein infrastruktureller Ansatz, mit dem Veränderungen der

Geschäftswelt dynamisch und flexibel begegnet werden können (vgl. Kamoun 2007, Übers. D. Verf.).

In der vorliegenden Arbeit wird kein Geschäftsprozess im betriebswirtschaftlichen Sinne betrachtet, sondern die Untersuchung bezieht sich auf Prozesse, die innerhalb der Berufsbildungsforschung stattfinden. Der Terminus „Geschäftsprozess“ soll dennoch beibehalten werden. Im Abschnitt 4.3 wurde bezüglich der ersten Fragestellung vereinbart, den Prozess der Lehrstellenzählung als berufswissenschaftlichen Kontext zu betrachten. Dieser bildet laut Krekel die Grundlage für die Ausbildungsplatzstellenbilanz im Berufsbildungsbericht (vgl. 2006, S. 206). An dieser Stelle soll jener Prozess unter Einhaltung der modellhaften Annahmen aus Abschnitt 4.3 rekonstruiert werden. Hierbei werden ausschließlich die fachlichen Anforderungen berücksichtigt. Des Weiteren wird nach der Bottom-Up-Methode vorgegangen, d. h., der Gesamtprozess wird sukzessive aus Teilprozessen zusammengesetzt (vgl. Stahlknecht / Hasenkamp 2005, S. 257).

Den Ausgangspunkt der Prozessanalyse bildet der Berufsbildungsbericht. Sein Entstehungsprozess geht zum Teil aus § 88 Nr. 4 Berufsbildungsgesetz (BBiG) hervor. Demnach werden die zu Zwecken der Erstellung des Berufsbildungsberichts sowie zur Durchführung der Berufsbildungsforschung erforderlichen Daten zum einen vom Bundesinstitut für Berufsbildungsforschung (BIBB) selbst und zum anderen vom Statistischen Bundesamt erfasst und an eine entsprechende Organisationseinheit des BIBB weitergeleitet (vgl. § 88 Abs. 4 BBiG). Folgt man darüber hinaus den Ausführungen von Althoff, kann der Prozess weiter vervollständigt werden (vgl. 2006, S. 202). Demzufolge beziehen die Statistischen Landesämter ihre Daten von den Kammern der gewerblichen Wirtschaft. Laut Krekel wird eine Form der Erhebung am 31.09. eines Kalenderjahres vom BIBB selbst in Auftrag gegeben (vgl. 2006, S. 207), wohingegen die Datenerhebung des Statistischen Bundesamtes am 31.12. aus § 88 BBiG hervorgeht. Beide Statistiken fließen in den Berufsbildungsbericht ein. Es ist außerdem bekannt, dass beide Statistiken zu ein und demselben Erhebungsgegenstand unterschiedliche Zahlen ausweisen (vgl. Krekel 2006, S. 208). Aus diesen Angaben werden folgende Geschäftsprozesse abgeleitet:

- Unternehmen mit entsprechender Eignung zur Ausbildung und einem Bedarf zur Nachwuchsgewinnung schreiben Ausbildungsstellen aus.
- Ausbildungsplatznachfrager/ -innen bewerben sich auf Ausbildungsstellen.
- Es erfolgt eine Eignungsprüfung in Form eines Bewerbungsgesprächs.

- Ausbildungsplatznachfrager/ -innen werden darüber informiert, ob sie den Ausbildungsplatz bekommen haben.
- Unternehmen und Ausbildungsplatznachfrager/ -innen schließen einen Ausbildungsvertrag.
- Der Ausbildungsvertrag wird von den Kammern der gewerblichen Wirtschaft erfasst.
- Sowohl das BIBB selbst als auch das Statistische Bundesamt führen zwei voneinander unabhängige Datenerhebungen durch.
- Die erhobenen Daten werden einer separaten Organisationseinheit des BIBB zur Erstellung des Berufsbildungsberichtes übermittelt.
- Im Berufsbildungsbericht werden die Informationen über die Ausbildungsmarktstellenbilanz veröffentlicht.

Die einzelnen Teilprozesse werden nun gemäß der Bottom-Up-Methode zu übergeordneten Prozessen zusammengefasst. Dazu wurden die Prozesskategorien „Ausschreibung der Ausbildungsstellenangebote“, „Bewerbungsverfahren“, „Entstehung eines Ausbildungsverhältnisses“ und „Bereitstellung der Daten“ gewählt. Wie in der Tabelle 1 zu sehen ist, bilden diese Teilprozesse den Gesamtprozess der Lehrstellenzählung.

Tabelle 1: Gesamtprozess der Lehrstellenzählung als Grundlage für die Entwicklung von Services.

| Lehrstellenzählung | | | |
|---|---|--|---|
| Ausschreibung der Ausbildungsstellenangebote | Bewerbungsverfahren | Entstehung eines Ausbildungsverhältnisses | Bereitstellung der Daten |
| Unternehmen mit entsprechender Eignung zur Ausbildung und einem Bedarf zur Nachwuchsgewinnung schreiben Ausbildungsstellen aus. | Ausbildungsplatznachfrager/ -innen bewerben sich auf Ausbildungsstellen. | Die Unternehmen und Ausbildungsplatznachfrager/ -innen schließen einen Ausbildungsvertrag. | Der Ausbildungsvertrag wird von den Kammern der gewerblichen Wirtschaft erfasst. |
| | Es erfolgt eine Eignungsprüfung in Form eines Bewerbungsgesprächs. | Der Ausbildungsvertrag wird von den Kammern der gewerblichen Wirtschaft erfasst. | Sowohl das BIBB selbst als auch das Statistische Bundesamt führen zwei voneinander unabhängige Datenerhebungen durch. |
| | Ausbildungsplatznachfrager/ -innen werden darüber informiert, ob sie den Ausbildungsplatz bekommen haben. | | Die erhobenen Daten werden einer separaten Organisationseinheit des BIBB zur Erstellung des Berufsbildungsberichtes übermittelt. Im Berufsbildungsbericht werden die Informationen über die Ausbildungsmarktenstellenbilanz veröffentlicht. |

Geschäftsprozesse, wie sie hier modelliert wurden, sind von einer konkreten Technologie unabhängig. Sie beziehen sich auf rein fachliche Funktionalitäten, die innerhalb der Lehrstellenanalyse erforderlich sind. Wie bereits ausgeführt, sollen diese Prozesse durch Informationstechnologien im Sinne von Services unterstützt werden. Dazu sind zwei Dinge erforderlich: Zum einen müssen die Prozesse ermittelt werden, in denen sich der Einsatz von Services eignet, zum anderen muss es einen zentralen Mechanismus geben, der Services zur Verfügung stellt. Diesem Aspekt widmet sich der nächste Abschnitt.

4.2.3 Entwurf

Um die Services von Dienst Anbietern so zu veröffentlichen, dass diese von allen Interessengruppen genutzt werden können, ist eine Systemarchitektur erforderlich. Nasr, Gross und Van Deursen (2011) kommen in ihren Studien zu dem Ergebnis, dass sich der Einsatz eines Web-Portals eignet, um verschiedene Service-Portfolios zu systematisieren und zentral zu verwalten. Darüber hinaus stellen sie fest, dass Web-Portale ein geeignetes Instrument darstellen, um unkundigen Interessengruppen die Funktion einer SOA aufzuzeigen (vgl. ebd.). Begriffe wie „Integration Center of Competence“ (ICoC) oder „System Integration Organisation“ (SIO) werden in diesem Zusammenhang von den untersuchten Unternehmen verwendet, um die Funktion der zentralen Verwaltungseinheit in Form eines Web-Portals zu verdeutlichen (vgl. ebd.). Auch in dem hier verfolgten Migrationskonzept soll dieses Verfahren zum Einsatz kommen. Dazu soll ein Web-Portal mit der Bezeichnung „Research Exchange Service“ (RES) entworfen werden, das die zentrale Verwaltung der Dienste im Sinne eines ICoC bzw. SIO übernimmt und zudem ein Service-Portfolio bereitstellt.

Es wird hierbei in drei Schritten vorgegangen: Zuerst wird das Zusammenwirken der einzelnen Elemente innerhalb der SOA beschrieben. Der Fokus richtet sich dabei auf die zentrale Verwaltungseinheit RES. Das Zusammenwirken der einzelnen Elemente soll einerseits aus einer technischen und andererseits aus einer funktionalen Sichtweise beschrieben werden. Anschließend erfolgt die Identifikation der relevanten Dienste auf der Grundlage der zuvor durchgeführten Geschäftsprozessanalyse. Des Weiteren werden die Adressinformationen und Dienstbeschreibungen der einzelnen Dienste in das Web-Portal eingefügt, sodass systematisch ein Service-Portfolio aufgebaut wird. Schließlich besteht das Ziel darin, diejenigen Dienste aus dem Service-Portfolio zu entnehmen, die für eine Lehrstellenzählung und -analyse verwendet werden können.

4.2.3.1 Architektur

In der Informatik hat sich laut Richling der Begriff „Architektur“ als Bezeichnung für den Aufbau und die Struktur von Hardware oder Software etabliert (vgl. 2011, S. 9). Eine Architektur definiert seinen Ausführungen zufolge Regeln, nach denen die einzelnen Elemente eines Systems interagieren. In Bezug auf die SOA lässt sich die Interaktion der einzelnen Elemente am Schema von Seth, Agarwal und Raj Singla verdeutlichen (vgl. 2011, S. 1-7, Übers. D. Verf.). Abbildung 3 zeigt, wie nach diesem Schema die Elemente Service-Anbieter, -nachfrager und Verzeichnisdienst miteinander interagieren. Schill und Springer erweitern dieses Schema, in dem sie den funktionalen Zusammenhang zwischen den einzelnen Elementen beschreiben (vgl. 2012, S. 20). Sie sind der Meinung, dass die zugrundeliegende Architektur eine prozessorientierte Sicht mit Diensten als Basiskonzept verfolgt, welche in verteilten Systemen angeboten, gesucht und genutzt werden können. Die Koordination zwischen Dienstanbieter und -nutzer übernimmt ihren Ausführungen zufolge die zentrale Verwaltungseinheit. Diese bildet den Kern der SOA und soll hier Basis des Systementwurfs sein.

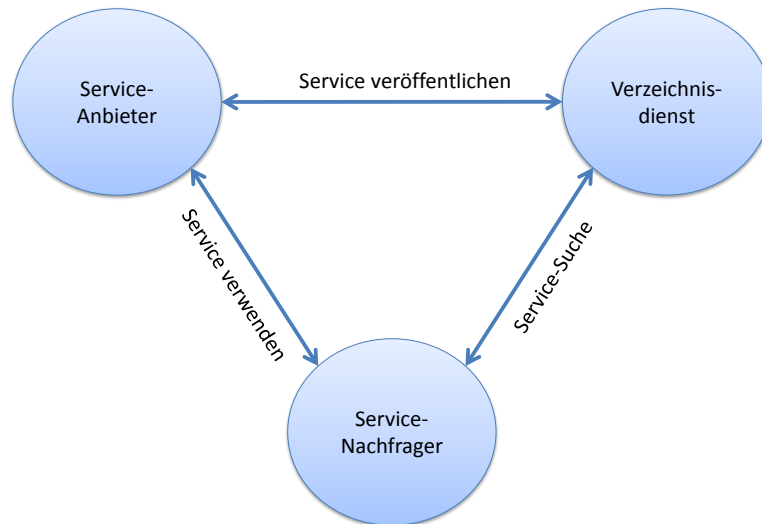


Abbildung 3. Kernelemente einer SOA nach Seth, Agarwal und Raj Singla (2011), erweitert um die funktionalen Beziehungen zwischen den einzelnen Elementen auf der Grundlage von Schill und Springer (2012, S. 20).

Folgt man diesem Grundschemata, so lässt sich das Web-Portal RES in die Kategorie Verzeichnisdienst einordnen. Wird allerdings ein Web-Portal als Verzeichnisdienst eingesetzt, müssen die Dienste bzw. Services über das Internet erreichbar sein. Hierzu haben sich Webdienste bzw. Webservices zu einer relativ eigenständigen und weit verbreiteten Technologie entwickelt (vgl. Dumke et al. 2003, S. 154). Das World Wide Web Consortium versteht unter einem Webservice ein Softwaresystem, das über eine URI erreichbar ist und von anderen Softwaresystemen eingebunden und wiederverwendet werden kann (vgl. W3C 2004, Übers. D. Verf.). Um allerdings das Anbieten und Einbinden mit dieser Technologie zu ermöglichen, sind Standards notwendig, die sich anhand des folgenden Schichtenmodells verdeutlichen lassen.

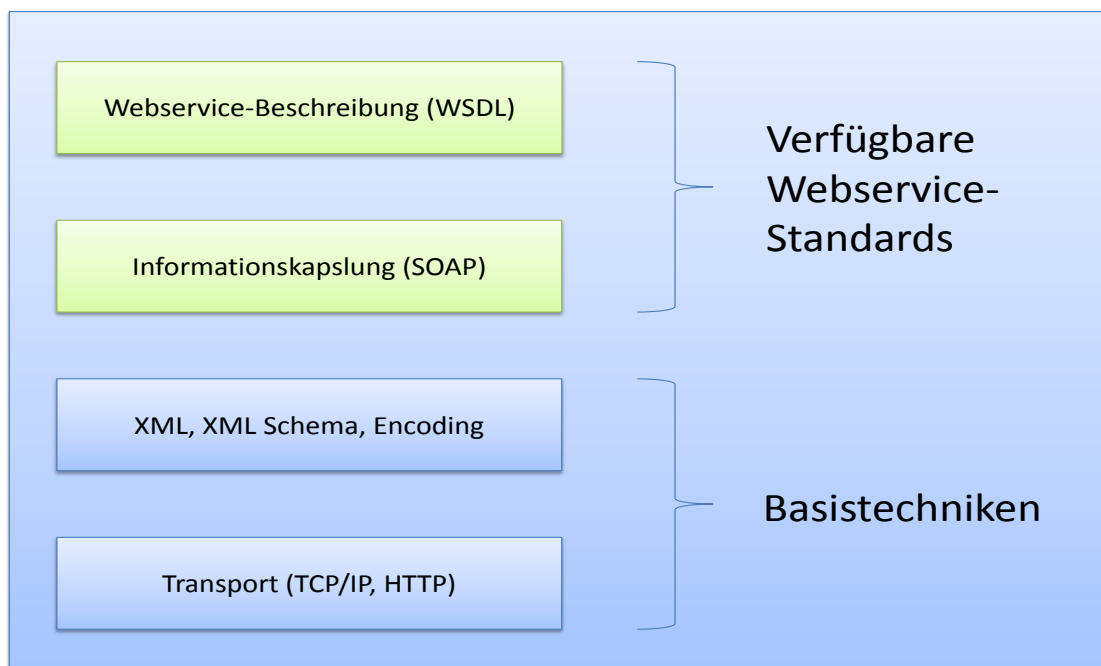


Abbildung 4: Webservice-Technologien im Überblick nach Dumke et al. (2003, S. 154).

Auf der oberen Ebene befindet sich die Web Service Description Language (WSDL). Sie wird vom Dienstanbieter verwendet, um die Funktion und Implementierung des Webservice für den Dienstanbieter zu beschreiben. Die Beschreibung lässt sich in zwei Bereiche aufteilen: einen abstrakten und einen konkreten. Während der abstrakte Bereich den Service hinsichtlich der verwendeten Datentypen und die darauf anzuwendenden Operationen beschreibt, werden im konkreten Teil Möglichkeiten vereinbart, um Daten bzw. Nachrichten zu transportieren. Überdies stellt der konkrete Teil Adressangaben für Dienstanbieter zur Verfügung. Diese werden benötigt, damit die Softwaresysteme der Dienstanfrage den Service finden und aufrufen können (vgl. W3C 2004, Übers. D. Verf.). Die Aufgabe der zentralen Verwaltungseinheit RES besteht darin, die Beschreibung der Webservices in Form einer XML-Datei bereitzustellen, sodass Dienstanbieter diese Datei über das Internet erreichen und innerhalb der eigenen Anwendung verwenden können. Die Datei nimmt dadurch die Funktion einer Schnittstelle ein, indem sie Dienstanbieter und -nachfrager miteinander verbindet. Um diese Schnittstelle realisieren zu können, muss die Verwaltungseinheit RES um eine Datenbank erweitert werden, in der die einzelnen Pfade der Webservices-Angebote persistent gespeichert werden können. Das Simple Object Access Protocol

(SOAP) stellt in diesem Zusammenhang das Bindeglied zwischen zwei heterogenen Systemen dar. Es beschreibt einerseits den Aufbau der zu übertragenden Nachricht und stellt andererseits ein standardisiertes Verfahren zur Verfügung, mit dem heterogene Systeme in der Lage sind, die Nachrichten des anderen Systems zu interpretieren. Der eigentliche Datentransport wird schließlich durch das Hyper-Transport-Protokoll realisiert (vgl. W3C 2004, Übers. D. Verf.).

Aus den bisherigen Beschreibungen geht hervor, dass die einzelnen Elemente der Architektur in Diensteanbietern, Dienstanfragern und Verwaltungseinheit bzw. Verzeichnisdienst unterschieden werden können. Auch sind die funktionalen Beziehungen zwischen den Elementen bekannt. Diese beziehen sich auf das Suchen, Bereitstellen und Einbinden von Webdiensten. Die zentrale Verwaltungseinheit RES stellt deshalb die im Folgenden aufgeführten Funktionen zur Verfügung:

- Zunächst wird eine Funktion bereitgestellt, mit der Diensteanbieter – in diesem Kontext sind es wissenschaftlich tätige Personen – ihren Dienst mittels einer WSDL-Datei inserieren können. Dies erfordert eine persistente Ablage der Dienstschnittstellen bzw. WSDL-Pfade.
- Des Weiteren wird eine Funktion benötigt, die es ermöglicht, Dienste zu suchen.
- Schließlich muss die zentrale Verwaltungseinheit RES die WSDL-Datei für jeden Dienst so über das Internet zur Verfügung stellen, dass sie von anderen Systemen problemlos erreicht und eingebunden werden können.

Selbstverständlich könnten diese Funktionalitäten an dieser Stelle mittels UML-Diagrammen konkreter beschrieben werden. Da jedoch das Migrationskonzept so abstrakt wie möglich bleiben soll, wird auf diese konkrete Beschreibungsform verzichtet. Ferner wurden die einzusetzenden Standards definiert, mit denen die SOA umgesetzt werden kann. Was in diesem Zusammenhang noch fehlt, ist eine Beschreibung der technologischen Infrastruktur, die es ermöglicht, dass die zentrale Verwaltungseinheit RES über das Internet erreichbar ist, Datenobjekte in einem Netzwerk gesendet, empfangen und darüber hinaus an geeigneter Stelle persistent gespeichert werden können. Hierzu ist ein Zusammenspiel verschiedener Technologien erforderlich.

In den Studien von Nasr, Gross und Van Deursen wird diese Form der Architektur als „Middleware“ bezeichnet (vgl. 2011, Übers. D. Verf.). Hierunter wird laut Stahlknecht und Hasenkamp eine spezielle Form der Unterstützungssoftware bezeichnet, die ein Verarbeiten und Übertragen von Daten in verteilten Systemen ermöglicht. Den Ausführungen von Hyun Jung und Soo Dong zufolge, wird im Kontext einer SOA dabei von einem „Enterprise-Service-Bus“ (ESB) gesprochen (vgl. 2011, S. 1288, Übers. D. Verf.). In dem hier betrachteten Fall wird ein Web-Portal realisiert, das eine zentrale Schnittstelle zwischen Dienst Anbietern und -nachfragern darstellt. Aus diesem Grund bilden ein Anwendungsserver und ein Datenbankserver den ESB. Die Abbildung 3 zeigt den ESB, welcher der hier verfolgten Migration zugrunde liegt.

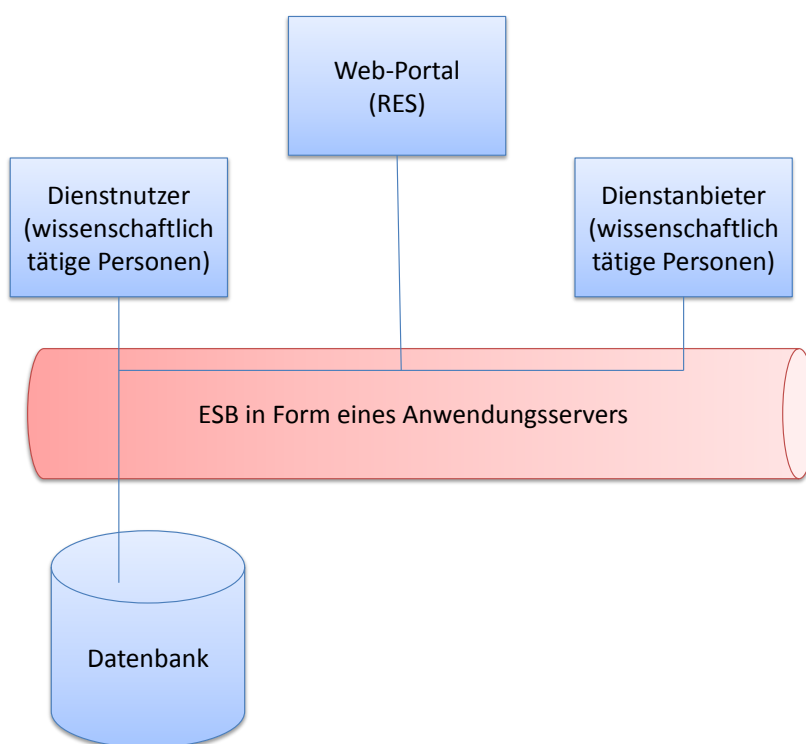


Abbildung 5: Enterprise-Service-Bus der serviceorientierten Forschungsdateninfrastruktur.

4.2.3.2 Datenspezifikation der Service-Anbieter

Damit die Dienstanbieter berufsbezogene Daten mit den Dienstutzern austauschen können, sind Datenobjekte erforderlich, die von jedem einzelnen System und jeder einzelnen Programmiersprache empfangen, verarbeitet und versendet werden können. Ziel dieses Abschnittes ist es, die für den Prozess der Lehrstellenzählung erforderlichen Datentypen, die mit ihnen durchführbaren Operationen sowie ihre Abhängigkeiten untereinander zu entwickeln. Um die Funktion von Datenobjekten zu verdeutlichen, ist es zunächst sinnvoll, Begriffe zu definieren, die im engen Zusammenhang mit Datenobjekten stehen. Hierzu zählen Datentypen und Datenstrukturen.

Computerprogramme verarbeiten verschiedene Sorten von Daten. Zahlen, Zahlen, Texte, Bilder, Musikstücke und Videos verbrauchen dabei nicht nur unterschiedlich viel Speicherplatz, mit ihnen sind außerdem unterschiedliche Operationen durchführbar. So lassen sich z. B. zwei Texte nicht addieren, wohl aber zwei Zahlen. Hieraus folgt, dass Daten einem bestimmten Typ entsprechen, der zugleich eine Menge charakteristischer Operationen bereitstellt. Programmiersprachen beinhalten standardmäßig eine Sammlung primitiver Datentypen samt zugehörigen Operationen (vgl. Gumm / Sommer 2009, S. 98). Ein primitiver Datentyp definiert für alle Daten seines Typs den zulässigen Wertebereich. Dieser kann entweder im Bereich der reellen Zahlen liegen, Wahrheitswerte umfassen oder aus einer endlichen Menge an Schriftzeichen bestehen (vgl. Stahlknecht / Hasenkamp 2005, S. 284). In Bezug auf die hier verfolgte Problemstellung lässt sich ein primitiver Datentyp wie folgt veranschaulichen: Für das Alter eines Auszubildenden sind natürlichen Zahlen zulässig, wohingegen der Name des Auszubildenden nur Zeichen eines vordefinierten Zeichensatzes enthalten darf. Diese Unterscheidung ist sinnvoll, da Informatiksysteme, wie oben gezeigt wurde, Zahlen anders verarbeiten als Schriftzeichen. Für eine genauere Betrachtung dieser Thematik sei an dieser Stelle auf Levi und Rembold verwiesen (vgl. 2003, S. 49-84).

In bestimmten Fällen reichen primitive Datentypen nicht aus. Wenn sich z. B. Daten aus einer Kombination unterschiedlicher Typen und Operationen zusammensetzen und diese zwischen verschiedenen Systemen und Programmiersprachen ausgetauscht werden sollen, ist die Entwicklung eines abstrakten Datentyps sinnvoll. Dabei wird der Typ so gestaltet, dass die Implementierung jederzeit geändert werden kann, ohne dass

die Verwertbarkeit des Typs innerhalb der anderen Systeme beeinträchtigt wird. Ein abstrakter Datentyp ist deshalb nur über eine Schnittstelle benutzbar. Die interne Realisierung bleibt verborgen (vgl. Saake / Sattler 2010, S. 259-260).

In der Literatur werden die Begriffe Datentyp und Datenstruktur synonym verwendet (vgl. Gumm / Sommer 2009, S. 98). In der vorliegenden Untersuchung wird an diesen Gebrauch der Begriffe angeknüpft. Eine Datenstruktur ist demnach eine Menge gleichartiger Daten, für die eine Sammlung von Operationen definiert ist (vgl. ebd.). Innerhalb der serviceorientierten Forschungsdateninfrastruktur sollen standardisierte abstrakte Datentypen zwischen verschiedenen heterogenen Systemen ausgetauscht werden. Aus diesem Grund widmet sich der folgende Abschnitt der Konstruktion dieser Datentypen.

Es geht zunächst um die Beantwortung der Frage, welche Objekte der realen Problemstellung im Informationssystem des Dienstanbieters dargestellt und dauerhaft (persistent) abgespeichert werden müssen. Da diese Daten die Grundlage für alle Funktionen und Abläufe bilden, die auf höherer Systemebene stattfinden, kommt der Modellierung ihrer Eigenschaften eine hohe Bedeutung zu (vgl. Hansen / Neumann 2009, S. 286).

Wie bereits erwähnt, besteht das Ziel der Datenmodellierung darin, das reale Problem – hier ist damit die Lehrstellenzählung gemeint – so präzise wie möglich einzugrenzen und abzubilden. Dabei kann das Problem nicht in seiner Gesamtheit erfasst werden, sondern nur ein als relevant erachteter Realweltausschnitt (vgl. ebd., S. 286). Im semantischen Datenmodell wird der Realweltausschnitt auf einer bestimmten Abstraktionsebene mithilfe einer grafischen Notation abgebildet. Unterscheidbare Objekte, die Bestandteil des Realweltausschnittes sind, werden dabei durch s.g. Entitäten repräsentiert. In diesem Modell stellen die Beziehungen der Entitäten eine logische Struktur aller Daten samt ihren Abhängigkeiten dar (vgl. Stahlknecht / Hasenkamp 2005, S. 163). Fehlt die logische Struktur, ist die Auswertung der Daten mit zusätzlichen Problemen verbunden. Als Folge können Anomalien und inkonsistente Datensätze auftreten, wodurch das Informationssystem in seiner Funktion, Informationen möglichst verzerrungsfrei abzubilden, eingeschränkt wird.

Weiterführende Informationen zum Thema Anomalien und Inkonsistenz können z. B. bei Ramez und Shamkant (2004, S. 499) nachgelesen werden.

Im Bereich der logischen Datenorganisation existieren neben semantischen Datenmodellen die relationalen Modelle (vgl. ebd., S. 163-178). Beide ergänzen sich zu einem Gesamtmodell: Während das semantische Modell eine grafische Notation verwendet, um die Kardinalitätsverhältnisse zwischen Entitäten abzubilden (vgl. Hansen / Neumann 2009, S. 288-289), werden im relationalen Modell alle Entitäten in zweidimensionale Tabellen umgewandelt und durch eine Relationsalgebra miteinander verbunden (vgl. Stahlknecht / Hasenkamp 2005, S. 172). Dem Relationsmodell liegt außerdem der Normalisierungsprozess zugrunde, bei dem schrittweise versucht wird, sogenannte Wiederholungsgruppen in Tabellen aufzuspüren und zu eliminieren. Wiederholungsgruppen sind gegeben, wenn mehrerer Werte innerhalb von Spalten derselben Zeile auftreten (vgl. ebd., S. 173). Es wurde nun geklärt, was unter den Begriffen Datentyp, Datenstruktur und Datenmodellierung verstanden wird. Ausgehend von diesem Erkenntnisstand wird nachfolgend ein Datenmodell entwickelt. Die Modellierung beginnt bei der Darstellung der persistenten, d. h. dauerhaften Datenspeicherung und geht dann in die von Programmiersprachen verwendeten Datenobjekte über.

Die folgende Datenmodellierung stützt sich auf das Modellierungsschema von Stahlknecht und Hasenkamp (vgl. 2009, S. 163-178). Demzufolge wird zunächst ein semantisches Datenmodell erarbeitet. Der Kernpunkt dieses Modells besteht den Autoren zufolge in der Beschreibung der logischen Struktur aller Daten. Dabei wird der Realweltausschnitt – damit ist hier die Lehrstellenzählung gemeint – samt der zugehörigen Realweltobjekte identifiziert und grafisch veranschaulicht. Als grafische Darstellungsform hat sich laut Stahlknecht und Hasenkamp das Entity-Relationship-Diagramm durchgesetzt. In diesem Diagramm werden die identifizierten Realweltobjekte als Entitäten bezeichnet. Als Basis für die Modellierung des Realweltausschnittes wurde § 88 BBiG gewählt. Dort legt der Gesetzgeber alle erhebungsrelevanten Merkmale fest und nennt implizit auch alle erforderlichen Realweltobjekte nebst ihren Abhängigkeiten. Im Rahmen dieser Untersuchung sind allerdings die sich aus der Abgrenzung des Forschungsgegenstandes ergebenden

Annahmen zu beachten. Die Tabelle 2 zeigt diesbezüglich eine Möglichkeit, um die im § 88 BBiG genannten Objekte des Realweltproblems zu identifizieren und zu strukturieren.

Tabelle 2: Identifizierung der Entitäten zur Entwicklung von standardisierten Datentypen.

| Realweltobjekt | Erhebungsmerkmal gem. § 88 BBiG | Attribute | Datentypen |
|--------------------|------------------------------------|--|---|
| Ausbildungsstellen | Abs. 1 Nr. 1. c), f) | Beginn, Ende, Abbruch, Auszubildender, Unternehmen ² , Ausbildungsberuf | Integer, Integer, Integer, Referenz auf Auszubildende, Referenz auf Unternehmen, Referenz auf Ausbildungsberufe |
| Unternehmen | Abs. 1 Nr. 1. d) | Ort, Wirtschaftszweig | String, Referenz auf Wirtschaftszweige |
| Auszubildende | Abs. 1 Nr. 1. a) | Geburtsjahr, Geschlecht, Staatsangehörigkeit | Integer, Integer, String |
| Ausbildungsberufe | Abs. 1 Nr. 1. c) | Bezeichnung | String |
| Wirtschaftszweige | Abs. 1 Nr. 1. d) | Bezeichnung | String |

Wie in der Tabelle 2 zu erkennen ist, bilden die Erhebungsmerkmale Ausbildungsberuf, Unternehmen und Wirtschaftszweig sowohl separate Entitäten als auch eigenständige Datentypen. Da Unternehmen mehrere Ausbildungsstellen ausschreiben, viele Unternehmen dem gleichen Wirtschaftszweig angehören und Auszubildende mehrere Ausbildungsplätze besetzen können, besteht eine binäre Beziehung zwischen diesen Erhebungsmerkmalen. Um diese binäre Beziehung beschreiben zu können wird das Entity-Relationship-Diagramm benötigt, das die sog. Kardinalitätsverhältnisse zwischen den Daten ausdrückt. Kardinalitätsverhältnisse bezeichnen laut Ramez und Shamkant die binären Beziehungen zwischen den Entitäten bzw. Realweltobjekten (vgl. 2004, S. 79). Dabei unterscheiden die Autoren zwischen drei Beziehungstypen:

² Unternehmen sind nicht im §88 BBiG als Erhebungsmerkmal angegeben. Sie wurden stellvertretend für das Merkmal Ausbildungsstätte aufgenommen.

- Bei einer 1:1-Beziehung bezieht sich eine Ausprägung eines Realweltobjektes auf genau eine Ausprägung eines anderen Realweltobjektes.
- Anders verhält es sich bei einer 1:*-Beziehung. Hier bezieht sich eine Ausprägung eines Realweltobjektes auf mehrere Ausprägungen eines anderen Realweltobjektes.
- Bei einer *:*-Beziehung beziehen sich die Ausprägungen eines Realweltobjektes auf mehrere Ausprägungen eines anderen Realweltobjektes.

Die Abbildung 6 stellt für die hier modellierte Lehrstellenzählung das Entity-Relationship-Diagramm dar.

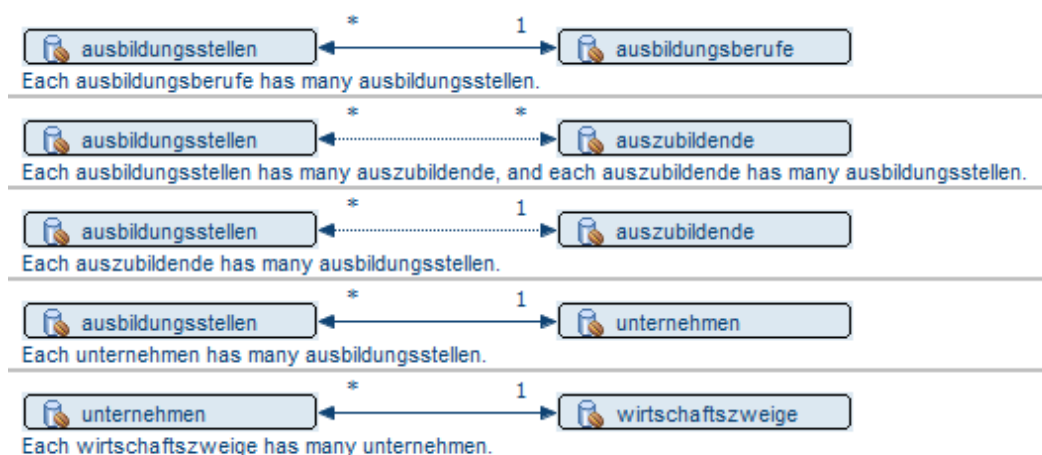


Abbildung 6: Semantisches Datenmodell, dargestellt durch das Entity-Relationship-Diagramm.

In einem zweiten Schritt wird das semantische Datenmodell in ein logisches Datenbankmodell umgewandelt. Laut Stahlknecht und Hasenkamp können dazu drei Modellierungstechniken in Betracht gezogen werden: das hierarchische Modell, das Netzwerkmodell oder das relationale Modell (vgl. 2009, S. 164). Für diese Untersuchung wurde dem relationalen Datenmodell der Vorzug gegeben, weil es sich in das hier eingesetzte Datenbankverwaltungssystem überführen lässt (vgl. ebd.). Genauere Informationen zum hier verwendeten Datenbankverwaltungssystem werden im Abschnitt 5.2.4 erläutert.

Im relationalen Datenbankmodell werden alle Daten in zweidimensionalen Tabellen mit einer festen Anzahl von Spalten und einer beliebigen Anzahl von Zeilen dargestellt. Ausgangspunkt bilden die zuvor ermittelten Entitäten. Wie in der Abbildung 7 zu erkennen ist, nehmen diese die Position der Tabelle ein. Die in der Tabelle 2

aufgeführten Attribute sind ebenfalls abgebildet. Diese bilden die Spalten der Tabellen. Eine bisher weitere Tabelle ist unter der Bezeichnung „servicestore“ zu finden. Diese bezieht sich nicht auf die Dienstanbieter, sondern auf das zentrale Verzeichnis, in dem die WSDL der Service-Angebote persistent gespeichert werden.

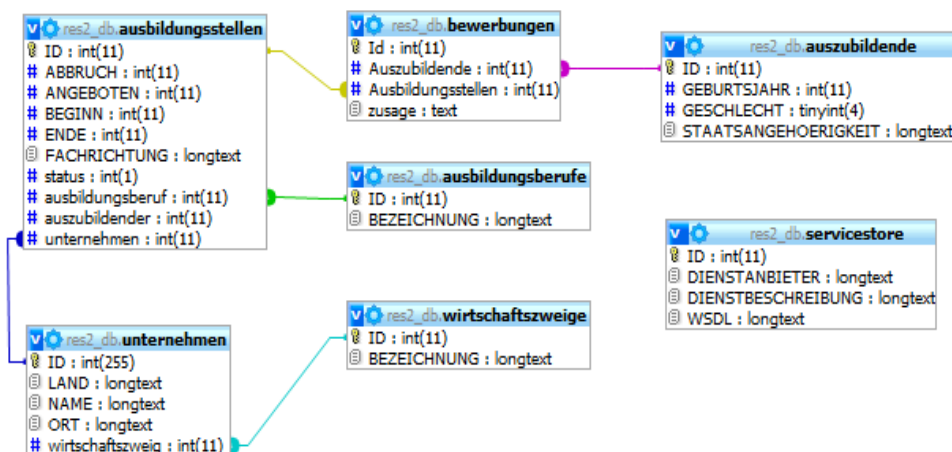


Abbildung 7: Das relationale Datenmodell zur persistenten Datenspeicherung.

Dem so entstandenen Datenmodell liegt ein zuvor durchgeführter Normalisierungsprozess zugrunde. Die im relationalen Datenmodell dargestellten Tabellen des Datenmodells – auch als Relationen bezeichnet – werden anschließend in ein Datenbankverwaltungssystem überführt. Damit stehen sie den Dienstanbietern zur dauerhaften Datenspeicherung zur Verfügung.

Bisher wurde ausschließlich die persistente Datenspeicherung in einem Datenbankverwaltungssystem betrachtet, ohne auf die Operationen einzugehen, mit denen die Datensätze innerhalb einer Programmiersprache bearbeitet werden können. Um das erarbeitete Datenmodell auch innerhalb von Programmiersprachen verwenden zu können, müssen Datenobjekte entwickelt werden, d. h. besondere Daten- und Programmstrukturen, die Eigenschaften und darauf definierte Operationen besitzen (vgl. Saake / Sattler 2010, S. 275). Saake und Sattler setzen den Begriff „Objekte“ gleich mit dem Begriff „abstrakte Datentypen“ (vgl. ebd.). Mit dieser Erkenntnis ist diese Untersuchung an dem Punkt angelangt, wo es um die Modellierung der abstrakten Datenobjekte geht, die innerhalb der serviceorientierten Forschungsdateninfrastruktur zwischen Dienstanbieter und Dienstnachfrager ausgetauscht werden sollen. Josuttis

(2008, S. 255) weist darauf hin, dass bei der Entwicklung von Datenobjekten, die innerhalb einer SOA zum Einsatz kommen, immer die richtige Balance zwischen Ausdrucksstärke und Komplexität gefunden werden muss (vgl. 2010, S. 255). Demnach sollten Datentypen so einfach wie möglich, aber nicht einfacher gestaltet werden. Dieses Prinzip wurde implizit bereits bei der Entwicklung des semantischen und logischen Datenmodells befolgt. Abbildung 8 beschreibt ein Klassenmodell, welches die Klassen der Datenobjekte darstellt. Eine Klasse ist nach Saake und Sattler (2010, S. 277) eine Menge von Objekten mit gleichen Eigenschaften und gleichem Verhalten. Bei genauerer Betrachtung weist diese Definition Ähnlichkeiten zu semantischen bzw. logischen Datenmodellen auf. Statt von Tabellen oder Entitäten wird im Klassendiagramm von Klassen ausgegangen. Der Unterschied besteht darin, dass Klassendiagramme in Programmiersprachen abgebildet werden können. Attribute werden dabei als Eigenschaften bezeichnet und Operationen als Methoden. Auf der Grundlage von Klassen können dann konkrete Datenobjekte erzeugt werden, die sich zwischen unterschiedlichen Systemen austauschen lassen. Abbildung 8 zeigt das Klassendiagramm der hier eingesetzten Datentypen samt den zugehörigen Abhängigkeiten.

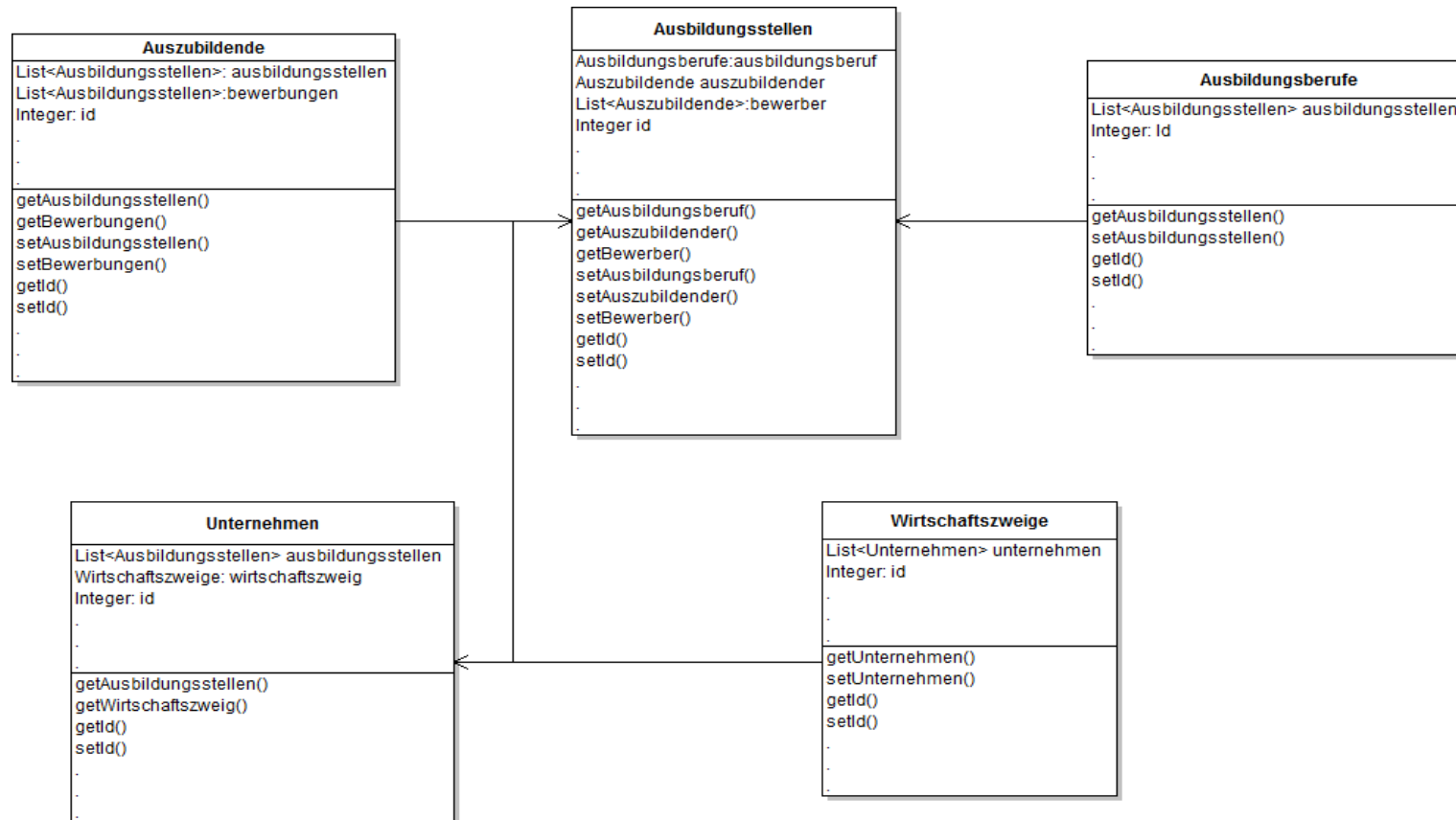


Abbildung 8: Die standardisierten Datentypen der Dienstanbieter.

4.2.3.3 Spezifikation der Services

Ausgehend von der Geschäftsprozessanalyse soll im Folgenden geklärt werden, in welchen Prozessen sich der Einsatz von Webservices eignet und welche Aufgabe dabei die Webservices übernehmen. Ziel ist die Entwicklung eines Service-Portfolios, das von den Diensteanbietern in die zentrale Verwaltungseinheit RES eingebunden und von unterschiedlichen wissenschaftlich tätigen Personen und Einrichtungen verwendet werden kann. Aus den Untersuchungen von Nasr, Gross und Van Deursen (2011, Übers. D. Verf.) geht hervor, dass zwei Möglichkeiten existieren, um den Einsatz von Services innerhalb eines Geschäftsprozesses zu identifizieren. Der empfohlene Weg ist dabei die Durchführung eines Workshops, an dem Vertreter der Interessengruppen beteiligt sind. Eine andere Möglichkeit besteht darin, das Einsatzgebiet eines Service innerhalb des Prozesses zu schätzen (vgl. ebd.). In dieser Arbeit wurde die zweite Variante gewählt. Obgleich sie im Gegensatz zur ersten nicht optimal ist, genügt diese Variante dennoch dem hier verfolgten Anspruch, ein exemplarisches Service-Portfolio aufzustellen.

Webservice zur Unternehmensauthentifizierung: Der Prozess der Lehrstellenanalyse beginnt mit der Lehrstellenausschreibung; die Ausschreibungen können anschließend in Form einer Datenliste und durch die Verwendung eines Webservice bezogen werden. Der erste Schritt besteht jedoch darin, dass sich das Unternehmen authentifizieren muss. Die Authentifizierung stellt sicher, dass nur diejenigen Unternehmen Ausbildungsstellen anbieten können, die fachlich und personell in der Lage sind, eine Ausbildung durchzuführen. Das setzt die Existenz einer Datenbank voraus, in der alle Unternehmen mit entsprechender Eignung eingetragen sind. In dem hier betrachteten Fall wird angenommen, dass eine solche Datenbank existiert. In der WSDL-Datei des Dienstes – hier der zentralen Verwaltungseinheit RES - wird eine Request-Response-Operation definiert. Bei dieser Operation sendet der Dienstanutzer dem Dienst eine Nachricht. Der Dienst empfängt diese Nachricht und sendet im Gegenzug eine Antwortet – die Unternehmen sind demnach die Dienstanutzer. Die von ihnen gesendeten Nachrichten enthalten einen Authentifizierungsschlüssel, dessen Datentyp aus Zeichen besteht. Ist der Authentifizierungsschlüssel korrekt, wird eine entsprechende Nachricht an das Unternehmen zurückgesendet. Die Nachricht enthält ein Datenobjekt vom Typ „Unternehmen“.

Webservice zur Identifizierung eines Ausbildungsberufes:

Ausbildungsstellenangebote beziehen sich auf einen konkreten Ausbildungsberuf. Da Ausbildungsberufe einem ständigen Wandel unterliegen, macht es Sinn, die aktuellen Ausbildungsberufe in Form eines Webservice zur Verfügung zu stellen. Dieser Webservice ermittelt alle verfügbaren Ausbildungsberufe. In diesem Fall wird eine Request-Response-Operation in der WSDL-Datei definiert. Wie im vorangegangenen Fall verdeutlicht wurde, sendet der Dienstanutzer dem Dienst eine Nachricht. Daraufhin versendet der Dienst eine Antwort an den Dienstanutzer, d. h. der Dienstanutzer sendet die Bezeichnung eines Ausbildungsberufes und erhält im Gegenzug das entsprechende Datenobjekt zum Ausbildungsberuf.

Webservice zur Registrierung von Ausbildungsstellen: Dieser Webservice registriert neue unbesetzte Ausbildungsstellen. Dazu benötigt der Dienst Informationen zur Ausbildungsstelle. In der WSDL-Datei wird deshalb eine One-Way-Operation definiert, d. h. der Dienstanutzer sendet eine Nachricht an den Dienst. Eine Antwort seitens des Dienstes erfolgt nicht. Mit der Nachricht werden einerseits die Identifikationsnummer des Ausbildungsberufes, andererseits die des Unternehmens übermittelt. Darüber hinaus beinhaltet die Nachricht Daten zum Ausbildungsbeginn, -abbruch, und -ende. Der Dienst erstellt daraufhin ein neues Datenobjekt und speichert es persistent ab.

Webservice zur Identifikation von unbesetzten Ausbildungsstellen: Der Webservice ermittelt alle unbesetzten Ausbildungsstellen eines Unternehmens. Dazu wird das Datenobjekt vom Typ Unternehmen an den Dienst gesendet. Der Dienst antwortet daraufhin mit einer Nachricht, die eine Liste aller diejenigen Datenobjekte enthält, die zum einen vom Typ Ausbildungsstellen sind und zum anderen dem Unternehmen zuzuordnen sind. Des Weiteren sind nur diejenigen Ausbildungsstellen enthalten, die vom Unternehmen als unbesetzt markiert sind. In der WSDL-Datei wird eine Request-Response-Operation definiert. Dieser Dienst lässt sich in beiden Kategorien einteilen: Einerseits kann er innerhalb der unternehmensinternen Systeme eingebunden und bspw. von der Personalabteilung verwendet werden und andererseits ist dieser Service auch für Dritte von Bedeutung. So ließe er sich in den Web-Portalen der Arbeitsagenturen einbinden, um die Stellenausschreibungen auf dem aktuellsten Stand zu halten, ohne dass eine manuelle Aktualisierung seitens der Portal-Administratoren notwendig ist.

Webservice zur Identifizierung von Bewerbern: Dieser Webservice ermittelt alle Bewerber, die sich auf eine bestimmte Stellenausschreibung beworben haben. Dazu wird ein Datenobjekt vom Typ Ausbildungsstellen an den Dienst gesendet. Dieser antwortet mit einer entsprechenden Nachricht, die eine Liste derjenigen Datenobjekte enthält, die vom Typ Auszubildende sind. Den vom Dienst gesendeten Datenobjekten ist ferner gemeinsam, dass sie eine n:m-Assoziation zur angegebenen Ausbildungsstelle aufweisen, wodurch eine Beziehung zwischen den Datenobjekten entsteht, die das Bewerbungsverhalten abbildet. In der WSDL-Datei wird eine Request-Response-Operation definiert.

Webservice zum Schließen eines Ausbildungsvertrages: Dieser Dienst schließt einen Ausbildungsvertrag zwischen einer Ausbildungsstelle und einem Bewerber/ einer Bewerberin. Dazu werden zwei Datenobjekte als Parameter an den Dienst übergeben. Zum einen ein Datenobjekt vom Typ Ausbildungsstellen und zum anderen ein Datenobjekt vom Typ Auszubildende. In der WSDL-Datei wird eine One-Way-Operation definiert, d. h. eine Antwort seitens des Dienstes erfolgt nicht.

Webservice zum Beziehen der Ausbildungsverträge: Mit diesem Dienst können Urlisten der abgeschlossene Ausbildungsverträge bezogen werden. Dabei werden die vollständigen Verträge in digitaler Form übermittelt, sodass durch den Einsatz entsprechender Analysesysteme die Daten hinsichtlich einer bestimmten Forschungsfrage frei kombiniert werden können. Der Dienst benötigt zunächst eine Jahreszahl. Auf Basis dieser Jahreszahl werden dann die entsprechenden Datenobjekte vom Typ Ausbildungsstellen übermittelt. In der WSDL-Datei wird eine Request-Response-Operation definiert.

Webservice zum Beziehen von Lehrstellenangebote: Dieser Service stellt die gesamten Lehrstellenangebote zur Verfügung, die von Unternehmen ausgeschrieben wurden. Auch hier wird die vollständige Stellenausschreibung inklusive aller Erhebungsmerkmale in elektronischer Form zur Verfügung gestellt. Dem Dienst wird eine Jahreszahl zugesendet. Anschließend antwortet der Dienst mit einer Liste der entsprechenden Lehrstellenausschreibungen. Hier wird ebenfalls wieder eine Request-Response-Operation definiert.

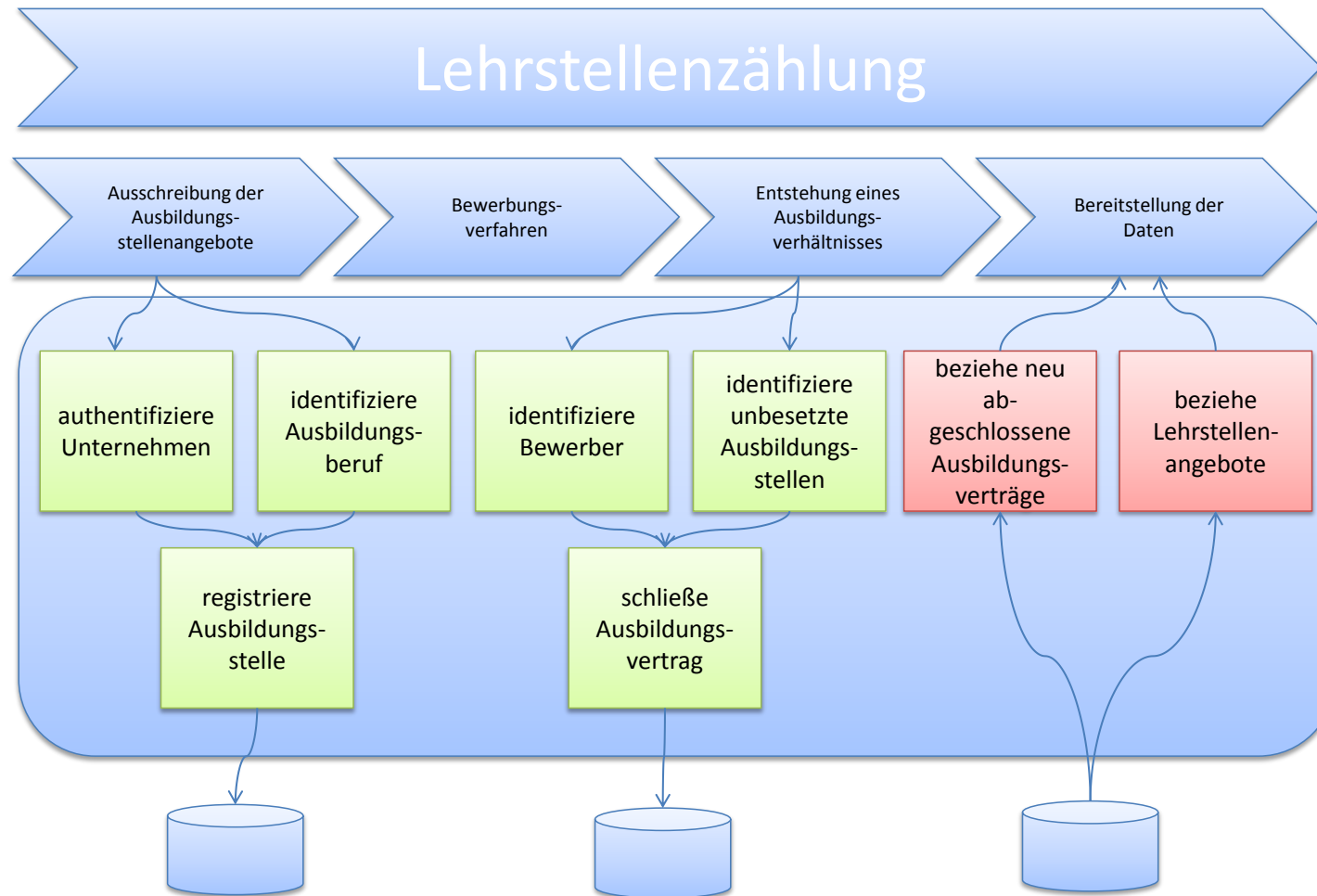


Abbildung 9: Grobentwurf einer serviceorientierte Forschungsdateninfrastruktur am Beispiel der Lehrstellenzählung. Erstellt auf der Grundlage der Geschäftsprozessanalyse; Services arbeiten choreografisch zusammen.

4.2.3.4 Zusammenfassung

Im vorangegangenen Abschnitt wurden drei Teilaspekte betrachtet: Zum einen wurde ein Architekturkonzept erarbeitet, mit dem es gelingen soll, die serviceorientierte Forschungsdateninfrastruktur umzusetzen. Dazu wurden die notwendigen technologischen Voraussetzungen und das funktionale Zusammenwirken der einzelnen Architekturelemente spezifiziert. Zum anderen wurden eine Reihe beispielhafter Dienste ermittelt. Allerdings stellen die hier ermittelten Dienste keine Standards dar. Vielmehr dienen sie als Beispiel, um die Machbarkeit einer serviceorientierten Forschungsdateninfrastruktur im Bereich der Berufsbildungsforschung zu veranschaulichen. Des Weiteren sind die Datenobjekte und Datentypen konstruiert worden, die zwischen den heterogenen Systemen ausgetauscht werden sollen. An dieser Stelle werden alle drei Aspekte zusammengeführt und mit der Notationssprache BPMN abgebildet. Das Modell wird in der Abbildung 10 dargestellt. Wie zu erkennen ist, folgt es demselben sequenziellen Ablauf wie die Geschäftsprozessanalyse aus Abschnitt 5.2.2. und der Service-Komposition aus Abschnitt 5.2.3.3. Der gesamte Rahmen bildet die serviceorientierte Forschungsdateninfrastruktur ab, bei der alle drei Teilnehmer miteinander in Verbindung stehen. Die Dienstanbieter werden in diesem Modell nicht dargestellt, d. h. es wird davon ausgegangen, dass die zuvor ermittelten Beispieldienste bereits über die zentrale Verwaltungseinheit RES bereitgestellt worden sind. Die oberste und die unterste Zeile (Lane) repräsentieren jeweils einen Teilnehmer, welcher am Prozess der Lehrstellenzählung und -analyse beteiligt ist. Der grün gefärbte Punkt am Anfang der ersten Zeile symbolisiert den Start des Geschäftsprozesses, wohingegen der rot gefärbte Punkt die Terminierung andeutet. Die gepunkteten Verbindungslinien stehen für eine Service-Operation. Darüber hinaus sind auch die zu übermittelnden Datenobjekte aus Abschnitt 5.2.3.2 dargestellt. Ob es sich um eine Input- oder Output-Operation handelt, ist an der grafischen Darstellung der Datenobjekte erkennbar. Während Datenobjekte mit einem ausgefüllten Pfeil in die rechte Richtung den Output des jeweiligen Service repräsentieren, stehen Datenobjekte, auf denen ein einfacher Pfeil abgebildet ist, für den Input. Die mittlere Zeile bildet die zentrale Verwaltungseinheit RES ab. Wie zu erkennen ist, enthält sie ein Service-Portfolio, bestehend aus den zuvor spezifizierten Services.

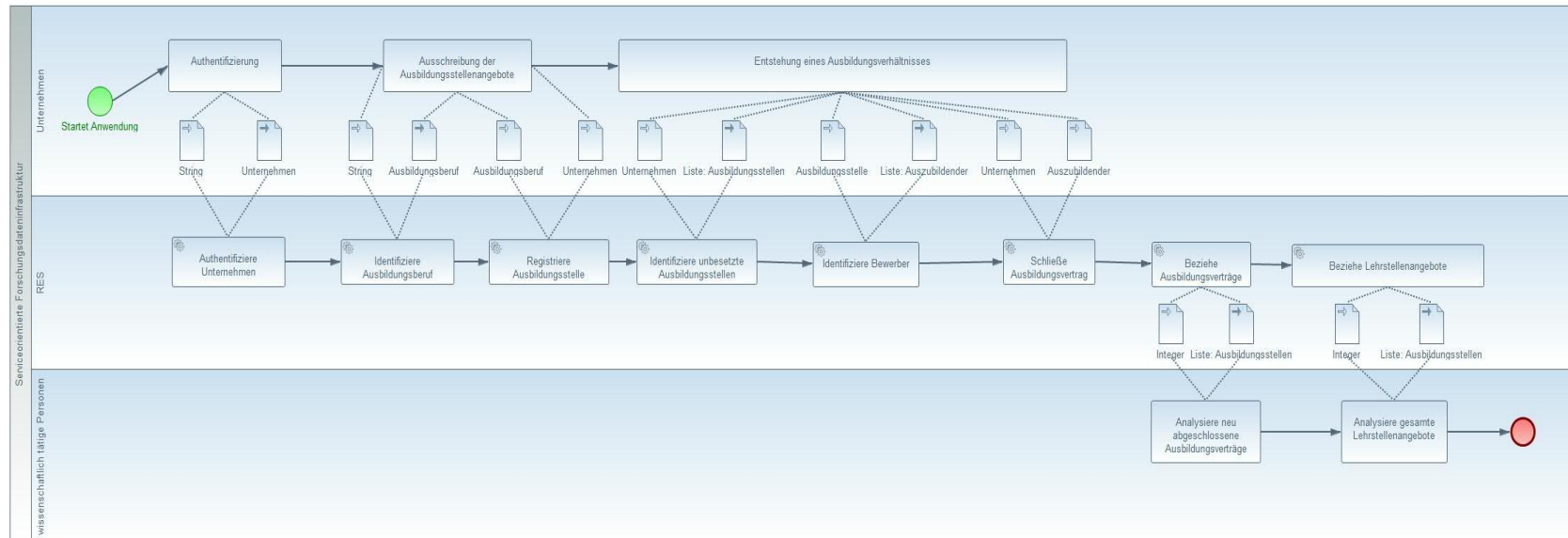


Abbildung 10: Feinentwurf einer serviceorientierte Forschungsdateninfrastruktur am Beispiel der Lehrstellenzählung und -analyse. Modelliert wurde mit der grafischen Notationssprache BPMN.

4.2.4 Realisierung

Das beschriebene Migrationskonzept wurde unter modellhaften Annahmen konstruiert und soll im Folgenden zu Simulationszwecken innerhalb einer Experimentierumgebung realisiert werden. Unter einer Experimentierumgebung wird hier ein Computersystem verstanden, das aus ausgewählten Hard- und Softwarekomponenten besteht. In diesem Fall wird das Modell T420 der Marke Lenovo als Zielsystem eingesetzt. Dieses System verfügt über einen Prozessor vom Typ Intel Core i5-2410M. Des Weiteren steht ein Arbeitsspeicher mit einer Kapazität von 3,98 GB zur Verfügung. Als Betriebssystem wird Windows in der Version 7 mit 64 Bit adressierbarem Arbeitsspeicher eingesetzt. Zur Implementierung des ESB dienen die von Nasr, Gross und Van Deursen identifizierten Basis-Technologien aus dem Java-Enterprise-Sektor (vgl. 2011, Übers. D. Verf.). Hierzu zählen in erster Linie ein auf der Programmiersprache Java basierender Anwendungsserver (JBoss AS 7) sowie entsprechende Programmierschnittstellen, die sowohl das Entwickeln von Webservices (JAX-WS-API) erleichtern als auch die persistente Datenspeicherung organisieren (JPA). Letzteres verweist auf den Einsatz eines relationalen Datenbankverwaltungssystems. Auf dem hier betrachteten Zielsystem wird dieser durch das Produkt MySQL in der Version 5.5.16 realisiert. Die genaue Implementierung der hier eingesetzten Datenbank entspricht dem relationalen Datenmodell aus Abschnitt 5.2.3.2. Um allerdings eine Datenbank innerhalb eines ESB verwenden zu können, sind entsprechende Serverkonfigurationen notwendig. Wie bereits erwähnt, bildet das Web-Portal RES die zentrale Verwaltungseinheit. Die Grundstruktur dieser Anwendung wurde mithilfe des Entwicklungswerkzeuges Maven 2 erstellt. Dabei handelt es sich um ein Hilfsmittel, das den Erstellungsprozess eines Anwendungsprogramms automatisiert. Das Entwicklungswerkzeug Maven 2 liefert darüber hinaus ein umfangreiches Repertoire an Entwurfsmustern, die eine saubere und nachvollziehbare Entwicklung des Web-Portals gewährleisten.

Nachdem die einzelnen Quelldateien der Dienstleister und des Web-Portals fertiggestellt worden waren, konnte die serviceorientierte Forschungsdateninfrastruktur in Betrieb genommen werden. Zuerst wurden der Anwendungsserver und das Datenbanksystem konfiguriert und gestartet. Die Konfigurationen können in der Anlage 10.2 nachvollzogen werden. Das Einsehen der Protokolle des Anwendungsservers

bestätigte, dass alle Konfigurationen korrekt verarbeitet worden waren. Anschließend wurden die Quelldateien des Web-Portals in ein für den Anwendungsserver lesbares Format („Web Application Archiv“) umgewandelt und auf dem Server eingespielt. Daraufhin war das Web-Portal unter der in der Abbildung 11 dargestellten Adresse erreichbar.

```
http://localhost:8080/RES/index.jsf
```

Abbildung 11: Die Adresse der zentralen Verwaltungseinheit RES auf dem Zielsystem.

Das Web-Portal wurde zuvor mit den hier dargestellten Webservices ausgestattet. Alle Webservices standen anschließend unter ihrer jeweiligen Adresse zur Verfügung. Die Adresse bezog sich auf die WSDL-Datei. Diese bildete die Schnittstelle zwischen Dienstanbieter und Dienstanwender. Um als Dienstanwender Zugriff auf die WSDL eines bestimmten Webservice zu bekommen, war es erforderlich, die in Abbildung 12 dargestellte URL in die Adresszeile des Webbrowsers einzugeben. Die Abbildung 12 zeigt den Aufruf des Webservice zur Identifikation von unbesetzten Ausbildungsstellen.


```
http://localhost:8080/RES/IdentifiziereUnbesetzteAusbildungsstellen?wsdl
```

Abbildung 12: Die WSDL-Adresse eines Webservice.

Anschließend ließen sich alle Dienste in die Systeme der Dienstanwender importieren. Zuvor wurden alle Systeme mit der Programmiersprache Java entwickelt. Die Programmiersprache stellte hierzu den in der Abbildung 13 dargestellten Befehl zur Verfügung. Dieser Befehl wird im Installationsverzeichnis der Java-Laufzeitumgebung ausgeführt und erzeugt daraufhin alle für den Dienstanwender erforderlichen Datentypen und Funktionen, die zuvor in der Entwurfsphase spezifiziert worden sind.

```
wsimport -s  
C:\Users\Name\Documents\workspace\Forschungsanwendung\src\main\java -p  
ovgu.ibbp.stub  
http://localhost:8080/RES/IdentifiziereUnbesetzteAusbildungsstellen?wsdl
```

Abbildung 13: Eine Möglichkeit zum Importieren von Webservices in Java.



(RES) Research-Exchange-Service

Themenbereich: Berufsbildungsforschung

Berufsbildungsforschung

Service-Portfolio zur Lehrstellenzählung

Neuen Service registrieren


Name des Services:

Anbieter:

WSDL:

Service-Portfolio:

| Bezeichnung | Dienstanbieter | WSDL |
|---|----------------|---|
| Service zur Authentifizierung | RES | http://localhost:8080/RES/AuthentifiziereUnternehmen?wsdl |
| Service zum Identifizieren der Ausbildungsberufe | BIBB | http://localhost:8080/RES/IdentifiziereAusbildungsberufe?wsdl |
| Service zur Registration von Ausbildungsstellen | RES | http://localhost:8080/RES/RegistriereAusbildungsstelle?wsdl |
| Service zur Identifikation von unbesetzten Ausbildungsstellen | RES | http://localhost:8080/RES/IdentifiziereUnbesetzteAusbildungsstellen?wsdl |
| Service zur Identifikation von Bewerbern | RES | http://localhost:8080/RES/IdentifiziereBewerber?wsdl |
| Service zur Identifikation von Bewerbern | RES | http://localhost:8080/RES/IdentifiziereBewerber?wsdl |
| Service zum Schließen von Ausbildungsverträgen | IHK | http://localhost:8080/RES/SchlieÙeAusbildungsvertrag?wsdl |
| Service um abgeschlossene Ausbildungsvertraege zu beziehen | BIBB | http://localhost:8080/RES/BezieheAusbildungsvertraege?wsdl |
| Service um Lehrstellenangebote zu beziehen | RES | http://localhost:8080/RES/BezieheLehrstellen?wsdl |

 JBoss Application Server 7

This project was generated from a Maven archetype from JBoss.

Abbildung 14: Benutzeroberfläche der zentralen Verwaltungseinheit RES mit Service-Portfolio zur Lehrstellenzählung.

4.3 Durchführung des Experiments

4.3.1 Vorbemerkung zum Experiment

Die Grundlage jeder Systemuntersuchung bildet zunächst eine verbale, umgangssprachliche Beschreibung des darzustellenden Sachverhaltes. Dabei ist eine klare Abgrenzung der Systemgrenzen, d. h. eine Abgrenzung zur Systemumwelt erforderlich. In den vorangegangenen Abschnitten wurde dieser Sachverhalt bereits ausführlich beschrieben. Es wurden sowohl die technologischen als auch die berufswissenschaftlichen Systemgrenzen konkret abgesteckt. In diesem Abschnitt soll das Migrationskonzept simuliert werden. Eine Simulation ist laut Roberts der Versuch, etwas zu imitieren (vgl. 1991, S. 3, Übers. D. Verf.). In dem hier betrachteten Fall werden computergestützte Erfassung- und Analysemethoden imitiert. Dabei wird eine deterministische, verhaltensklärende Simulation angestrebt. Diese Art der Simulation unterscheidet sich von klassischen stochastischen Simulationsmethoden. Während bei stochastischen Simulationen untersucht wird, welche statistische Verteilung des Verhaltens bei sich zufällig ändernden Einflussfaktoren zu erwarten ist, versuchen verhaltensklärende Simulationen die Verhaltensübereinstimmung zwischen System und Modell darzustellen.

In der Vorbereitungsphase zu diesem Experiment hat sich allerdings gezeigt, dass die Simulation von serviceorientierten Architekturen ein noch wenig erforschtes Gebiet darstellt. Dennoch belegen die Untersuchungen von Nasr, Gross und Van Deursen, dass in der Praxis entsprechende Methoden Anwendung finden, die Ähnlichkeiten zu einer verhaltensklärenden Simulation aufweisen (vgl. 2011, Übers. D. Verf.). Die Autoren verwenden dabei den Begriff „Simulation“ im Sinne von „testen“. In diesem Fall werden die einzelnen Webservices hinsichtlich ihrer funktionalen Spezifikation getestet (siehe Abschnitt 5.2.3.3). Um dies zu ermöglichen, stehen entsprechende Testwerkzeuge zur Verfügung. Der ebenfalls auf Java-Technologien basierende Anwendungsserver Glassfish v3 des Softwareherstellers Oracle ist bspw. ein solches Testwerkzeug. Bei genauerer Betrachtung hat sich allerdings gezeigt, dass beim Glassfish v3 nur solche Webservices getestet werden können, deren Input aus assoziationsfreien Datentypen besteht. Da im Rahmen dieser Arbeit abstrakte Datentypen mit Assoziationen gewählt wurden, entfällt diese Option. Stattdessen wird

eine eigene Simulationssteuerung programmiert, mit der die Instanzen der entsprechenden abstrakten bzw. komplexen Datentypen erzeugt und den Webservices übergeben werden können. Die Simulationssteuerung kann in den Anlagen eingesehen werden. Etwaige Fehlermeldungen werden abgefangen und ausgewertet. Schließlich folgt die Korrektur der Fehler. Dieses Verfahren wiederholt sich so lange, bis die Webservices gemäß der oben definierten Spezifikationen funktionieren. Aus diesem inkrementellen Testverfahren soll ein Machbarkeitsnachweis erbracht werden, der die Frage beantwortet, ob das dargestellte Migrationskonzept aus technischer Sicht realisiert werden kann. Außerdem dient es dem Aufspüren von möglichen Problembereichen.

4.3.2 Vorbereitung

In der Vorbereitungsphase des Testverfahrens wird eine Simulationssteuerung programmiert. Diese wird benötigt, um den Ausgangszustand des Modells herzustellen. Dazu erzeugt die Simulationssteuerung die Population der Ausbildungsplatznachfrager/-innen sowie jeweils ein Datenobjekt vom Typ Unternehmen, Ausbildungsberuf und Wirtschaftszweig. Die Datenobjekte werden anschließend von der Verarbeitungslogik persistent im Datenbanksystem abgespeichert. Während der Simulation werden alle Webservices hinsichtlich ihrer Spezifikation getestet. Dazu müssen die Programme der Dienstanwender exemplarisch nachgebaut werden. In dem hier betrachteten Fall werden zwei Programme benötigt. Das erste stellt eine beispielhafte Unternehmensanwendung dar. Diese bezieht von der zentralen Verwaltungseinheit RES diejenigen Dienste, die zum Erstellen eines Ausbildungsverhältnisses benötigt werden. Des Weiteren wird eine beispielhafte Analysesoftware programmiert. Diese dient wissenschaftlichen Personen, um auf die Datenbestände mittels Webservices zuzugreifen. Im Abschnitt 2 wurde gezeigt, dass SPSS, STATA, R und andere Softwareprodukte häufig zur quantitativen Datenanalyse verwendet werden. Hier wird allerdings als exemplarische Analysesoftware ein auf der Programmiersprache Java aufbauendes Programm eingesetzt. Dieses Programm bezieht diejenigen Dienste, die zur Analyse der Lehrstellenangebote sowie der neu abgeschlossenen Ausbildungsverträge notwendig sind.

4.3.3 Simulationsbeschreibung

Die Simulation verläuft sequenziell und orientiert sich entlang des Geschäftsprozesses, wie er im Abschnitt 5.2.3.3 beschrieben wurde. Sie beginnt mit der Definition der Ausgangssituation. Anschließend werden nacheinander die einzelnen Webservices durchlaufen und getestet. Dabei wird zwischen der Unternehmensanwendung, der Systemsteuerung und der Analysesoftware gewechselt. Das Testverfahren bezieht sich allerdings ausschließlich auf die Input- und Output-Parameter derjenigen Webservices, die entweder in der Unternehmensanwendung oder in der Analysesoftware eingebunden sind. Zur Aufzeichnung des Systemverhaltens wird das im Abschnitt 5.3.5 beschriebene Ereignisprotokoll verwendet.

4.3.4 Ziele der Simulation

Ein Ziel der Simulation besteht darin, die technische Funktionalität der serviceorientierten Forschungsdateninfrastruktur zu überprüfen. In diesem Zusammenhang sollen Informationen darüber gewonnen werden, ob die entwickelten Webservices innerhalb der modellhaften serviceorientierten Forschungsinfrastruktur ihr zuvor spezifiziertes Verhalten zeigen. Dazu werden die Input-Parameter konstant gehalten und die zurückgegebenen Werte der Dienste ausgewertet. Doch nicht nur die technischen Aspekte sollen in der Simulation berücksichtigt werden, sondern auch die Verwendbarkeit der Dienste für wissenschaftlich tätige Personen. Um darüber eine Aussage treffen zu können, werden die Webservices zur Analyse der Lehrstellenangebote und der neu abgeschlossenen Ausbildungsverträge getestet. Dabei ist zu hinterfragen, ob diese

Dienste den Datennutzern tatsächlich die Möglichkeit bieten, die Urlisten der Datenerhebungen über das Internet bzw. der zentralen Verwaltungsstelle RES zu beziehen und ob sich diese Urlisten automatisch zwischen Datenlieferant und Datennutzer synchronisieren. In diesem Kontext wäre mit Blick auf die gesamte Architektur zu hinterfragen, ob den wissenschaftlich tätigen Personen die Funktion bereitgestellt wird, eigene Datenerhebungen über einen Dienst bereitzustellen und Dienste anderer Anbieter zu integrieren. In Bezug auf die gesamte Architektur wäre weiterhin zu fragen, inwiefern hierdurch ein europäischer Forschungsraum entstehen kann. Nicht alle diese Fragen können aus den Ergebnissen der Simulation abgeleitet

werden. Allerdings stellt die hier entworfene modellhafte serviceorientierte Forschungsinfrastruktur einen über die Simulation hinausgehenden Interpretationsansatz bereit.

4.3.5 Testverfahren und -protokoll

Zur Aufzeichnung des Systemverhaltens wird das Ereignisprotokoll aus Tabelle 3 verwendet. Kommt es während der Simulation zu einem Fehler, wird dieser in den Log-Dateien des ESB bzw. des Anwendungsservers vermerkt. Dieser Vermerk wird anschließend in das Ereignisprotokoll übertragen. In der Spalte Operation wird der Operationstyp angegeben. Dieser wurde für jeden Webservice spezifiziert und kann im Abschnitt 5.2.3.3 eingesehen werden. Die Spalte „Input“ bezieht sich auf die Parameter, die der Dienst zur Abarbeitung seiner Aufgabe entgegennimmt. Demzufolge bezieht sich die Spalte „Output“ auf die vom Dienst generierte Antwort. In der Spalte „Status“ können die Werte „O.K.“ für die korrekte Verhaltensweise des Webservices oder „ERROR“ für ein mögliches Fehlverhalten eingetragen werden. Insgesamt werden zwei Testdurchläufe durchgeführt. Die Differenz der Ereignisprotokolle soll Auskunft über mögliche Probleme und deren Lösungsmöglichkeiten im Rahmen des hier erarbeiteten Migrationskonzeptes aufzeigen.

Tabelle 3: Der Aufbau des Ereignisprotokolls.

| Testdurchlauf | | | | | |
|----------------------|-----------|-------|--------|--------|-----------------|
| Service | Operation | Input | Output | Status | Fehlermeldungen |
| | | | | | |
| | | | | | |

4.3.6 Testergebnisse

Tabelle 4: Das Ergebnisprotokoll des 1. Testdurchlaufes.

| Testdurchlauf 1 | | | | | |
|--------------------------------|----------------------------|---|----------------------------------|--------|-----------------|
| Service | Operation | Input | Output | Status | Fehlermeldungen |
| authentifiziere Unternehmen | Request-Response-Operation | Objekt vom Typ String | Objekt vom Typ Unternehmen | O.K. | 0 |
| identifiziere Ausbildungsberuf | Request-Response-Operation | Objekt vom Typ String | Objekt vom Typ Ausbildungsberufe | O.K. | 0 |
| registriere Ausbildungsstelle | One-Way-Operation | Objekt vom Typ Ausbildungsberufe/ Objekt vom Typ Unternehmen | nicht definiert | O.K. | 0 |

| Testdurchlauf 1 | | | | | |
|---|----------------------------|-----------------------------------|---|--------|--|
| Service | Operation | Input | Output | Status | Fehlermeldungen |
| identifiziere unbesetzte Ausbildungsstellen | Request-Response-Operation | Objekt vom Typ Unternehmen | Listenobjekt vom Typ Ausbildungsstellen | ERROR | Marshalling Error: failed to lazily initialize a collection of role: ovgu.ibbp.model.Ausbildungsstellen.bewerber, no session or session was closed |
| identifiziere Bewerber | Request-Response-Operation | Objekt vom Typ Ausbildungsstellen | Listenobjekt vom Typ Auszubildende | ERROR | Failed to access the WSDL at: http://localhost:8080/RES/ServiceCollection?wsdl . It failed with: Got Connection refused: connect while opening stream from http://localhost:8080/RES/ServiceCollection?wsdl . |
| schließe Ausbildungsvertrag | One-Way-Operation | Objekt vom Typ Ausbildungsstellen | Objekt vom Typ Auszubildende | ERROR | Failed to access the WSDL at: http://localhost:8080/RES/ServiceCollection?wsdl . It failed with: Got Connection refused: connect while opening stream from http://localhost:8080/RES/ServiceCollection?wsdl . |

| Testdurchlauf 1 | | | | | |
|---------------------------------------|------------------------------------|---------------------------|--|--------|---|
| Service | Operation | Input | Output | Status | Fehlermeldungen |
| beziehe Aus- bildungs- verträge | Request- Response- Operation | Objekt vom Typ Integer | Listenobjekt vom Typ Ausbildungsstellen | ERROR | Failed to access the WSDL at: http://localhost:8080/RES/ServiceCollection?wsdl. It failed with: Got Connection refused: connect while opening stream from http://localhost:8080/RES/ServiceCollection?wsdl. |
| beziehe Lehr- stellen- angebote | Request- Response- Operation | Objekt vom Typ Integer | Listenobjekt vom Typ Ausbildungsstellen | ERROR | Failed to access the WSDL at: http://localhost:8080/RES/ServiceCollection?wsdl. It failed with: Got Connection refused: connect while opening stream from http://localhost:8080/RES/ServiceCollection?wsdl. |

Tabelle 5: Das Ergebnisprotokoll des 2. Testdurchlaufes.

| Testdurchlauf 2 | | | | | |
|---|----------------------------|---|---|--------|-----------------|
| Service | Operation | Input | Output | Status | Fehlermeldungen |
| authentifiziere Unternehmen | Request-Response-Operation | Objekt vom Typ String | Objekt vom Typ Unternehmen | O.K. | 0 |
| identifiziere Ausbildungsberuf | Request-Response-Operation | Objekt vom Typ String | Objekt vom Typ Ausbildungsberufe | O.K. | 0 |
| registriere Ausbildungsstelle | One-Way-Operation | Objekt vom Typ Ausbildungsberufe/ Objekt vom Typ Unternehmen | nicht definiert | O.K. | 0 |
| identifiziere unbesetzte Ausbildungsstellen | Request-Response-Operation | Objekt vom Typ Unternehmen | Listenobjekt vom Typ Ausbildungsstellen | O.K. | 0 |

| Testdurchlauf 2 | | | | | |
|-----------------------------|----------------------------|-----------------------------------|---|--------|-----------------|
| Service | Operation | Input | Output | Status | Fehlermeldungen |
| identifiziere Bewerber | Request-Response-Operation | Objekt vom Typ Ausbildungsstellen | Listenobjekt vom Typ Auszubildende | O.K. | 0 |
| schließe Ausbildungsvertrag | One-Way-Operation | Objekt vom Typ Ausbildungsstellen | Objekt vom Typ Auszubildende | O.K. | 0 |
| beziehe Ausbildungsverträge | Request-Response-Operation | Objekt vom Typ Integer | Listenobjekt vom Typ Ausbildungsstellen | O.K. | 0 |
| beziehe Lehrstellenangebote | Request-Response-Operation | Objekt vom Typ Integer | Listenobjekt vom Typ Ausbildungsstellen | O.K. | 0 |

4.3.7 Auswertung

Das hier entwickelte Migrationskonzept zielte darauf ab, eine serviceorientierte Forschungsdateninfrastruktur innerhalb der Berufsbildungsforschung zu realisieren. Dazu wurde die Lehrstellenzählung als Bestandteil der Berufsbildungsforschung unter modellhaften Annahmen rekonstruiert und als exemplarisches Beispiel eingesetzt. Um die Validität des Konzeptes zu prüfen, wurde das Migrationskonzept in einer verhaltensklärenden Simulation getestet. Im Folgenden sollen die Ergebnisse des Testverfahrens ausgewertet werden.

4.3.7.1 Technologische Funktionalität

Hinsichtlich der technologischen Funktionalität kann zunächst festgehalten werden, dass alle Dienste ohne große Umstände über das Web-Portal bezogen und in die Endsysteme der Dienstnutzer integriert werden konnten. Dazu mussten zuerst Anwendungs- und Datenbankserver gestartet werden. Das Einsehen der Ereignisprotokolldateien des Anwendungsservers bestätigte, dass alle unter Anlage 10.2 dargestellten Konfigurationen problemlos abgearbeitet werden konnten. Beim Einbinden der Webservices in die Desktopanwendungen der Dienstnutzer offenbarten sich vor allem die Vorteile der im Migrationskonzept eingesetzten Standardtechnologien: Durch die Verwendung der bereits anerkannten Standards wie WSDL und SOAP können Webservices in nahezu allen geläufigen Entwicklungsumgebungen und Analysesoftwaressystemen eingebunden werden, da die Hersteller sich in der Regel an diese Standards halten und Integrationswerkzeuge zur Verfügung stellen.

Nach der Integration der Webservices in die Desktop-Anwendungen der Dienstnutzer wurde die Simulation eingeleitet. Sequenziell wurden alle Webservices getestet. Dabei kam es im ersten Testdurchlauf während der Identifizierung der unbesetzten Ausbildungsstellen zu einem Fehler, der einen Ausfall des Web-Portals verursachte.

Dieses Fehlverhalten ist einerseits auf die im Migrationskonzept entwickelten Datentypen und andererseits auf das Übertragungsprotokoll SOAP zurückzuführen, denn die Beziehungen und Abhängigkeiten zwischen den einzelnen Datenobjekten konnten nicht in eine SOAP-Nachricht umgewandelt und mittels HTTP übertragen werden. Das Migrationskonzept in der Ausgangsfassung erwies sich deshalb als nicht umsetzungsfähig. Erst das Deaktivieren der Beziehungen zwischen den Datenobjekten bewirkte im zweiten Testdurchlauf, dass alle Webservices funktionierten.

4.3.7.2 Anwendungsbezogene Funktionalität

Für die Präsentation und die Analyse der berufsbezogenen Daten mittels computergestützter Analysemethoden werden die letzten beiden Dienste benötigt. Zudem wurde der Versuch unternommen, die Dienste jeweils in dem Tabellenkalkulationsprogramm Microsoft Excel – in einer auf der Programmiersprache Java aufbauenden Web-Anwendung sowie in verschiedenen, ebenfalls auf der Programmiersprache Java basierenden Desktop-Anwendungen zu integrieren. Dies ist in allen drei Fällen problemlos gelungen. Allerdings ist diese Tatsache nicht zuletzt den Softwareprodukten und Entwicklungsumgebungen selbst zu verdanken, da sie bereits vordefinierte Funktionen zur Verfügung stellen, mit denen das Einbinden von Webservices realisiert werden kann. Obwohl das Integrieren der Analyse-Dienste keine erwähnenswerten Umstände bereitete, stellte sich heraus, dass in allen drei Fällen Programmierkenntnisse erforderlich waren, um die Dienste innerhalb der Anwendungen anzusprechen. So zeigte sich beispielsweise, dass ein Diagramm, welches die neu abgeschlossenen Ausbildungsverträge getrennt nach männlichen und weiblichen Auszubildenden darstellen sollte, nur durch die Einbeziehung der Programmiersprache Visual Basic in Excel realisiert werden konnte. Hier wäre zu hinterfragen, ob wissenschaftlich tätige Personen im Bereich der Berufsbildungsforschung über diese Kenntnisse in ausreichendem Maße verfügen und ob diese Notwendigkeit der Flexibilität einer serviceorientierten Forschungsdateninfrastruktur entgegenwirkt. Positiv zu erwähnen ist, dass sich durch den Einsatz von Programmiersprachen und SOAP-Nachrichten alle Datensätze hinsichtlich einer beliebigen Merkmalskombination analysieren lassen.

5 Schlussfolgerung

Die Simulation des Migrationskonzepts am Modell zeigt, wie eine serviceorientierte Forschungsdateninfrastruktur im Bereich der Berufsbildungsforschung umgesetzt werden könnte. Es stellt sich nun die Frage, ob das hier dargestellte Konzept eine Mängelminimierung bezüglich der angesprochenen Defizite aktueller Informationssysteme leisten kann. Um die Frage beantworten zu können, werden nicht nur die Ergebnisse der Simulation in Betracht gezogen, sondern auch die Erfahrungen, die während der Entwicklungsphase und Inbetriebnahme gesammelt wurden.

Das erste Problem betrifft die BIBB-FDZ sowie die elektronischen Datenbanken der statistischen Ämter. Es äußert sich darin, dass einmal erworbene Datenbestände auf den Systemen der Datennutzer nicht automatisch aktualisiert werden: Sobald sich die Datensätze erweitern, besitzen die Datennutzer einen veralteten Datenbestand. Diesem Problem kann mit einer serviceorientierten Forschungsdateninfrastruktur entgegengewirkt werden. So zeigte sich, dass bei den Dienstnutzern, die den Service zum Beziehen der Lehrstellenangebote verwendeten, die angeforderten Daten nach dem Simulationsdurchlauf in aktualisierter Form vorlagen. Allerdings setzt eine solche Aktualisierung eine bestehende Internetverbindung voraus.

Als weiteres Problem wurde der fehlende Austausch von Forschungsergebnissen zwischen wissenschaftlich tätigen Personen angesprochen. Zurzeit tritt allein das BIBB als Dienstanbieter in Erscheinung und bestimmt somit das Angebot an Untersuchungsdaten. Mit einem Web-Portal als zentraler Verwaltungsstelle können allerdings nicht nur Forschungseinrichtungen wie das BIBB oder die statistischen Ämter, forschungsrelevante Daten zur Verfügung stellen, sondern auch einzelne wissenschaftlich tätige Personen. Wie in der Realisierungsphase gezeigt wurde, müssen dazu die WSDL-Dateien der Forschungsdatenanbieter im Web-Portal registriert werden. Hierdurch werden Forschungsergebnisse transparenter, wodurch gleichzeitig auch die Wiederverwendung einmal erhobener Daten steigt. Hierdurch ist auf den ersten Blick eine hohe Vernetzung zwischen Forschungsprogrammen möglich, die sich auf den gesamten europäischen Forschungsraum ausdehnen könnte.

Des Weiteren besteht das Ausgangsproblem, dass die Datensätze der statistischen Ämter in geschlossenen Strukturen angeboten werden, wodurch eine freie

Merkmalskombination hinsichtlich einer bestimmten Forschungsfrage nicht möglich ist. Dieses Problem besteht bei einer serviceorientierten Architektur nicht, weil die Daten mittels SOAP übertragen werden. Dabei handelt es sich um ein Dokumentenformat, dessen Inhalte in hierarchischen Strukturen gegliedert sind. Entsprechende Computerprogramme können beliebig auf einzelne Inhaltselemente zugreifen, wodurch eine freie Merkmalskombination gewährleistet und folglich ein hohes Maß an Flexibilität bei der Analyse von Forschungsdaten gegeben ist. Allerdings geht diese Flexibilität mit einem Zuwachs an erforderlichen Programmierkenntnissen einher. Im Rahmen dieser Untersuchung entstand deshalb der Eindruck, dass serviceorientierte Forschungsdateninfrastrukturen nach dem hier entwickelten Konzept besser für große Organisationen geeignet sind, bei denen der Programmieraufwand vom Fachpersonal übernommen werden kann.

Besonders gut lässt sich diese Annahme anhand eines Beispiels verdeutlichen. So ist zu vermuten, dass die Bundesagentur für Arbeit ein Interesse daran hat, Dienste, mit denen unbesetzte Ausbildungsstellen verschiedener Unternehmen ermittelt werden können, in die eigenen Systeme zu implementieren. Mit diesen Diensten könnte die Agentur für Arbeit ihre Stellenausschreibungen immer auf dem aktuellen Stand halten, ohne Investitionen in Aktualisierungsmaßnahmen tätigen zu müssen; denn Organisationen von dieser Größe verfügen häufig über eigene IT-Abteilungen, in denen Fachkräfte die notwendigen Programmertätigkeiten ausführen. In diesem Sinne wäre auch der Fall denkbar, dass Fachkräfte im Auftrag der Organisation eigene Dienste über das Web-Portal zur Verfügung stellen. In der Simulation zeigte sich darüber hinaus, dass das SOAP nur bedingt geeignet ist, Nachrichten mit komplexem Inhalt zu übertragen. Ferner stellte sich heraus, dass der Einsatz einer zentralen Verwaltungseinheit in Form eines Web-Portals nach dem hier entwickelten Konzept nicht optimal ist. So führte das Fehlverhalten eines Webservice gleichzeitig zu einem Ausfall des Web-Portals, wodurch auch die anderen Webservices nicht mehr zur Verfügung standen.

In dieser Untersuchung konnte außerdem festgestellt werden, dass die Bereitstellung eigener Dienste nach dem hier beschriebenen Konzept einen hohen Entwicklungsaufwand erfordert. So müssen z.B. Datentypen, WSDL-Dateien und Algorithmen zur Verarbeitung der Daten von den Diensteanbietern selbst entwickelt werden. Hier wäre zu hinterfragen, ob wissenschaftlich tätige Personen aus dem

Fachbereich der Berufsbildungsforschung über die dafür notwendigen Kenntnisse verfügen. Dies wurde bereits weiter oben thematisiert. Aber auch als Dienstanutzer sind Programmierkenntnisse notwendig. So ist zwar das Einbinden der Dienste in vielen Softwareprodukten über eine grafische Benutzerschnittstelle gewährleistet, dennoch ist die anschließende Verwendung der Dienste nur mit einer Programmiersprache möglich. Insofern könnte sich der hohe Entwicklungsaufwand nachteilig auf die interoperable Nutzung von berufsbezogenen Forschungsdaten auswirken.

Ein weiterer, bisher noch nicht betrachteter Aspekt betrifft den Datenschutz. Im Gegensatz zur Geodateninfrastruktur beziehen sich berufsbezogene Daten auf Individuen, deren persönliche Daten laut § 5 IFG vor Missbrauch geschützt werden müssen. In dem hier betrachteten Migrationskonzept wurde angenommen, dass Informationen über Auszubildende und Unternehmen in einer Datenbank persistent abgespeichert sind. Gemäß § 28 Abs. 4 BDSG ist eine Verarbeitung oder Nutzung personenbezogener Daten unzulässig, wenn der Betroffene dieser Nutzung widerspricht. Angesichts dieser gesetzlichen Regelung darf man begründete Zweifel daran haben, dass eine freie Bereitstellung personenbezogener Daten für wissenschaftliche Analysezwecke gelingen kann. Auch diese Tatsache könnte sich nachteilig auf eine europaweite Vernetzung von Forschungsprogrammen im Sinne einer serviceorientierten Forschungsdateninfrastruktur auswirken.

Unter Abwägung aller Vor- und Nachteile kann zusammenfassend festgehalten werden, dass sich das hier dargestellte Modell einer serviceorientierten Forschungsdateninfrastruktur nicht im vollen Umfang als Informationssystem im Bereich der Berufsbildungsforschung eignet. Auf der einen Seite werden zwar die angesprochenen Mängel aktueller Informationssysteme minimiert, auf der anderen Seite hätte eine Migration eine Reihe neuer Probleme zur Folge. Hier wäre der mit der Lösung der Probleme verbundene Aufwand im Hinblick auf das Verhältnis zu dem angesprochenen Nutzen erst noch zu untersuchen.

6 Empfehlungen für weiterführende Forschungsansätze

Mit dem hier erarbeiteten und erprobten Migrationskonzept sind eine Menge Erkenntnisse entstanden, die den Überförungsprozess zu einer serviceorientierten Forschungsdateninfrastruktur im Bereich der Berufsbildungsforschung vorantreiben können. Die in dieser Untersuchung aufgeworfenen Schwierigkeiten bilden dabei eine wichtige Grundlage für weiterführende Forschungsansätze.

Zunächst muss das Entwickeln von Webservices auch für diejenigen wissenschaftlich tätigen Personen zugänglich gemacht werden, die nicht explizit über Programmierkenntnisse verfügen. Hierzu sollten entsprechende Softwarewerkzeuge entwickelt werden, die den Entwicklungs- und Bereitstellungsprozess von forschungsrelevanten Daten und Service-Funktionen automatisieren.

Wie am Anfang der Untersuchung beschrieben, wird außerdem eine effizientere Nutzung der berufswissenschaftlichen Forschungsdaten angestrebt. Im Hinblick auf die Effizienz wäre zu untersuchen, welcher Allokationsmechanismus die Prämisse gewährleistet, dass forschungsrelevante Daten und Softwaredienste dort eingesetzt werden, wo sie den größtmöglichen Nutzen stiften. Im Rahmen dieser Arbeit wurde dafür bereits die Grundlage in Form eines Web-Portals geschaffen. Hierdurch begegnen sich Anbieter und Nachfrager. Unbeantwortet bleibt zunächst noch die Frage, wodurch die Qualität der bereitgestellten Daten und Dienste abgesichert werden kann, so dass ein mögliches Marktversagen vermieden wird. In diesem Zusammenhang muss auch die Frage geklärt werden, wer die Rechte an den Forschungsdaten und Dienstangeboten besitzt.

Auch auf der rechtlichen und organisatorischen Ebene müssen entsprechende Untersuchungen stattfinden. Vor allem das Thema „Datenschutz“ bedarf einer ausführlichen Prüfung. So müssen Möglichkeiten und Gesetzesgrundlagen zur Anonymisierung der forschungsrelevanten Daten geschaffen werden, die sich speziell an die Anforderungen einer serviceorientierten Forschungsdateninfrastruktur im Bereich der Berufsbildungsforschung richten.

Literatur

- Althoff, H. (2006): Berufsbildungsstatistik. In: Rauner, F. (Hrsg.): Handbuch Berufsbildungsforschung. 2. Aufl., Bielefeld: Bertelsmann Verlag, S. 199-204.
- Balzert, H. (2001): Lehrbuch der Software-Technik. 2. Aufl. Heidelberg: Spektrum Akademischer Verlag.
- Boehm, B. W. (1987): A spiral model of software development and enhancement. In: Computer, 21, 5, S. 61-72.
- Brügge, B. / Dutoit, A. H. (2004): Objektorientierte Softwaretechnik mit UML, Entwurfsmustern und Java. München, Boston: Pearson Studium.
- Bundesministerium für Bildung und Forschung (BMBF) (2005): Berufsbildungsgesetz. BBiG. URL: <<http://www.bmbf.de/pubRD/bbig.pdf>>, zuletzt geprüft: 29.06.2012
- Chamoni, P. / Gluchowski P. (2006): Analytische Informationssysteme – Einordnung und Überblick. In: Chamoni P. / Gluchowski P. (2006) (Hrsg.): Data Warehouse, On-Line Analytical Processing, Data Mining. Berlin Heidelberg: Springer Verlag, S. 3-25.
- Dumke, R. / Lothar, M. / Wille, C. / Zbrog, F. (2003): Web Engineering. München: Pearson-Studium.
- Europäischer Rat (2000): Schlussfolgerungen des Vorsitzes. Lissabon, URL: <http://www.europarl.europa.eu/summits/lis1_de.htm>, zuletzt geprüft: 29.06.2012.
- Europäischer Rat (2007): Schlussfolgerungen des Vorsitzes. Brüssel, URL: <<http://register.consilium.europa.eu/pdf/de/07/st07/st07224-re01.de07.pdf>>, zuletzt geprüft: 04.06.2012.
- Europäischer Rat (2011): Schlussfolgerungen des Rates zur Rolle der allgemeinen und beruflichen Bildung bei der Durchführung der Strategie Europa 2020. URL: <<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:C:2011:070:0001:0003:DE:PDF>>, zuletzt geprüft: 07.06.2012.
- Friedrich, A. / Hohn C. / Rohrbach-Schmidt D. (2011): BIBB-FDZ Daten- und Methodenberichte Nr. 1 / 2011. Ergebnisse der BIBB-FDZ-Onlineumfrage. URL: <<http://www.bibb.de/bibb-fdz-nutzerumfrage>>, zuletzt aktualisiert am 15.02.2012, zuletzt geprüft: 25.06.2012.

Geodatenzugangsgesetz – GeoZG (2009). URL: <<http://www.gesetze-im-internet.de/bundesrecht/geozg/gesamt.pdf>>, zuletzt geprüft: 28.06.2012.

Gumm, H.P. / Sommer M. (2009): Einführung in die Informatik. 8. Aufl. München, Wien: Oldenbourg.

Hansen, H. R. / Neumann, G. (2009): Wirtschaftsinformatik 1. Grundlagen und Anwendungen. 10. Aufl., Stuttgart: Lucius & Lucius Verlag.

Hyun Jung, L. / Soo Dong, K. (2011): Static and dynamic adaptations for service-based systems. In: Information and Software Technology 53, 12, S. 1275-1296.

Josuttis, N. (2008): SOA in der Praxis. 1. Aufl., Heidelberg: dpunkt-Verlag.

Kamoun, F. (2007): A Roadmap towards the Convergence of Business Process Management and Service Oriented Architecture. In: Ubiquity, 2007, Nr. 2, URL: <<http://ubiquity.acm.org/article.cfm?id=1247273>>, zuletzt geprüft: 29.06.2012.

Krekel, E. M. (2006): Berufsbildungsbericht. In: Rauner, F. (Hrsg.): Handbuch Berufsbildungsforschung. 2. Aufl., Bielefeld: Bertelsmann Verlag, S. 204-208.

Levi, P. / Rembold, U. (2003): Einführung in die Informatik für Naturwissenschaftler und Ingenieure. 4. Aufl., München / Wien: Hanser Verlag.

Nasr, K. A. / Gross, H.-G. / Van Deursen, A. (2011): Realizing service migration in industry-lessons learned. In: Journal of Software Maintenance and Evolution: Research and Practice. DOI: 10.1002/SMR.540 URL: <<http://onlinelibrary.wiley.com/store/10.1002/smr.540/asset/540 ftp.pdf?v=1&t=h3fwtqvh&s=fa39bcae9def24288014a7eef099204184140082>>, zuletzt geprüft: 14.06.2012.

Ramez, E. / Shamkant, N. (2004): Grundlagen von Datenbanksystemen. 3. Aufl. München / Boston: Pearson Studium.

Rauner, F. (2006): Berufsbildungsforschung - Eine Einführung. In: Rauner, F. (Hrsg.): Handbuch Berufsbildungsforschung. 2. Aufl. Bielefeld: Bertelsmann Verlag, S. 9-16.

Rechenberg, P. / Pomberger, G. (2002): Informatik-Handbuch. 3. Aufl., München: Hanser Fachbuch Verlag.

Richling, J. (2011): Autonomie in verteilten IT-Architekturen. München: Oldenbourg.

Roberts, N. (1983): Introduction to computer simulation. The system dynamics approach. Mass: Addison-Wesley.

Saake, G. / Sattler, K.-U. (2010): Algorithmen und Datenstrukturen. Eine Einführung mit Java. 4. Aufl., Heidelberg: dpunkt Verlag

Schill, A. / Springer, T. (2012): Verteilte Systeme. Grundlagen und Basistechnologien. 2. Aufl. Berlin: Springer Verlag.

Seth, A. / Agarwal, H. / Raj Singla, A. (2011): Designing a SOA Based Model. In: ACM SIGSOFT Software Engineering Notes 36, 5, S. 1-7.

Stahlknecht, P. / Hasenkamp, U. (2005): Einführung in die Wirtschaftsinformatik. 11. Aufl. Berlin, New York: Springer Verlag.

Uhly, A. / Flemming, S. / Schmidt, D. / Schüller, F. (2009): Zwei Erhebungen zu neu abgeschlossenen Ausbildungsverträgen. Konzeptionelle Unterschiede zwischen der „Berufsbildungsstatistik zum 31.12.“ und der „BIBB-Erhebung über neu abgeschlossene Ausbildungsverträge zum 30.09.“. URL: <https://www.destatis.de/DE/Methoden/Methodenpapiere/Download/MethodenpapierBIBBB_StBA.pdf?__blob=publicationFile, zuletzt aktualisiert am 23.04.2009>, zuletzt geprüft: 25.06.2012.

Verein Deutscher Ingenieure (2000): Richtlinie 3633. Simulation von Logistik-, Materialfluß und Produktionssystemen. URL: <http://www.tu-ilmeneau.de/fileadmin/media/fabrikbetrieb/ayman/vdi3633_1.pdf>, zuletzt geprüft: 29.06.2012.

W3C (2004): Web Services Architecture. URL: <<http://www.w3.org/TR/ws-arch/>>, zuletzt geprüft: 30.06.2012

Zuser, W. / Grechenig, T. / Köhle, M. (2004): Software Engineering. Mit UML und dem Unified Process. 2. Aufl. München / Boston: Pearson Studium

Anlagen

Datentypen

Auszubildende.java

```
package ovgu.ibbp.model;

import java.io.Serializable;

/**
 * The persistent class for the auszubildende database table.
 */
@Entity
public class Auszubildende implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private int id;

    private int geburtsjahr;

    private byte geschlecht;

    @Lob()
    private String staatsangehoerigkeit;

    //bi-directional many-to-many association to Ausbildungsstellen
    @ManyToMany(mappedBy="bewerber", cascade=CascadeType.REMOVE)
    private List<Ausbildungsstellen> bewerbungen;

    //bi-directional many-to-one association to Ausbildungsstellen
    @OneToMany(mappedBy="auszubildende", cascade=CascadeType.REMOVE)
    private List<Ausbildungsstellen> ausbildungsstellen;

    public Auszubildende() {
    }

    public int getId() {
        return this.id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public int getGeburtsjahr() {
        return this.geburtsjahr;
    }

    public void setGeburtsjahr(int geburtsjahr) {
        this.geburtsjahr = geburtsjahr;
    }

    public byte getGeschlecht() {
        return this.geschlecht;
    }

    public void setGeschlecht(byte geschlecht) {
        this.geschlecht = geschlecht;
    }

    public String getStaatsangehoerigkeit() {
        return this.staatsangehoerigkeit;
    }
}
```

```
Ausbildungsberufe.java

package ovgu.ibbp.model;

import java.io.Serializable;

/**
 * The persistent class for the ausbildungsberufe database table.
 *
 */
@Entity
public class Ausbildungsberufe implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private int id;

    @Lob()
    private String bezeichnung;

    //bi-directional many-to-one association to Ausbildungsstellen
    @OneToMany(mappedBy="ausbildungsberufe", cascade=CascadeType.REMOVE, fetch=FetchType.EAGER)
    private List<Ausbildungsstellen> ausbildungsstellens;

    public Ausbildungsberufe() {
    }

    public int getId() {
        return this.id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getBezeichnung() {
        return this.bezeichnung;
    }

    public void setBezeichnung(String bezeichnung) {
        this.bezeichnung = bezeichnung;
    }

    @XmlTransient
    public List<Ausbildungsstellen> getAusbildungsstellens() {
        return this.ausbildungsstellens;
    }

    public void setAusbildungsstellens(List<Ausbildungsstellen> ausbildungsstellens) {
        this.ausbildungsstellens = ausbildungsstellens;
    }
}
}
```

```
Ausbildungsstellen.java

package ovgu.ibbp.model;

import java.io.Serializable;

/**
 * The persistent class for the ausbildungsstellen database table.
 *
 */
@Entity
public class Ausbildungsstellen implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private int id;

    private Integer abbruch;

    private int angeboten;

    //bi-directional many-to-one association to Auszubildende
    @ManyToOne
    @JoinColumn(name="auszubildender")
    private Auszubildende auszubildende;

    private Integer beginn;

    private int ende;

    @Lob()
    private String fachrichtung;

    private int status;

    //bi-directional many-to-one association to Ausbildungsberufe
    @ManyToOne(cascade=CascadeType.REMOVE)
    @JoinColumn(name="ausbildungsberuf")
    private Ausbildungsberufe ausbildungsberufe;

    //bi-directional many-to-one association to Unternehmen
    @ManyToOne(cascade=CascadeType.REMOVE)
    @JoinColumn(name="unternehmen")
    private Unternehmen unternehmen;

    //bi-directional many-to-many association to Auszubildende
    @ManyToMany(cascade=CascadeType.REMOVE)
    @JoinTable(
        name="bewerbungen"
        , joinColumns={
            @JoinColumn(name="Ausbildungsstellen")
        }
        , inverseJoinColumns={
            @JoinColumn(name="Auszubildende")
        }
    )
    private List<Auszubildende> bewerber;

    public Ausbildungsstellen() {
    }

    public int getId() {

```

Unternehmen.java

```
package ovgu.ibbp.model;

import java.io.Serializable;

/**
 * The persistent class for the unternehmen database table.
 */
@Entity
public class Unternehmen implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private int id;

    @Lob()
    private String land;

    @Lob()
    private String name;

    @Lob()
    private String ort;

    //bi-directional many-to-one association to Ausbildunesstellen
    @OneToMany(mappedBy="unternehmen", cascade=CascadeType.REMOVE)
    private List<Ausbildungsstellen> ausbildungsstellens;

    //bi-directional many-to-one association to Wirtschaftszweige
    @ManyToOne(cascade=CascadeType.REMOVE)
    @JoinColumn(name="wirtschaftszweig")
    private Wirtschaftszweige wirtschaftszweige;

    public Unternehmen() {
    }

    public int getId() {
        return this.id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getLand() {
        return this.land;
    }

    public void setLand(String land) {
        this.land = land;
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

```
Wirtschaftszweige.java

package ovgu.ibbp.model;

import java.io.Serializable;

/**
 * The persistent class for the wirtschaftszweige database table.
 *
 */
@Entity
public class Wirtschaftszweige implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private int id;

    @Lob()
    private String bezeichnung;

    //bi-directional many-to-one association to Unternehmen
    @OneToMany(mappedBy="wirtschaftszweige", cascade=CascadeType.REMOVE)
    private List<Unternehmens> unternehmens;

    public Wirtschaftszweige() {
    }

    public int getId() {
        return this.id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getBezeichnung() {
        return this.bezeichnung;
    }

    public void setBezeichnung(String bezeichnung) {
        this.bezeichnung = bezeichnung;
    }

    @XmlTransient
    public List<Unternehmens> getUnternehmens() {
        return this.unternehmens;
    }

    public void setUnternehmens(List<Unternehmens> unternehmens) {
        this.unternehmens = unternehmens;
    }
}
}
```

Serverkonfiguration

```
standalone.xml
88     <subsystem xmlns="urn:jboss:domain:datasources:1.0">
89         <datasources>
90             <datasource jndi-name="java:/res2_db" pool-name="MySQLDSRES2" enabled="true"
jta="true" use-java-context="true" use-ccm="true">
91                 <connection-url>
92                     jdbc:mysql://localhost:3306/res2_db
93                 </connection-url>
94                 <driver>
95                     com.mysql
96                 </driver>
97                 <transaction-isolation>
98                     TRANSACTION_READ_COMMITTED
99                 </transaction-isolation>
100                <pool>
101                    <prefill>
102                        false
103                    </prefill>
104                    <use-strict-min>
105                        false
106                    </use-strict-min>
107                    <flush-strategy>
108                        FailingConnectionOnly
109                    </flush-strategy>
110                </pool>
111                <security>
112                    <user-name>
113                        root
114                    </user-name>
115                </security>
116            </datasource>
117            <drivers>
118                <driver name="com.mysql" module="com.mysql">
119                    <xa-datasource-class>
120                        com.mysql.jdbc.jdbc2.optional.MysqlXADataSource
121                    </xa-datasource-class>
122                </driver>
123            </drivers>
124        </datasources>
125    </subsystem>
```

Zusammengefasste WSDL-Datei

ServiceCollection.xml

```
1 <?xml version='1.0' encoding='UTF-8'?><wsdl:definitions name="ServiceCollectionService"
2 targetNamespace="http://service.ibbp.ovgu/"
3 xmlns:ns1="http://schemas.xmlsoap.org/soap/http"
4 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://service.ibbp.ovgu/"
5 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
6 <wsdl:types>
7 <xs:schema elementFormDefault="unqualified" targetNamespace="http://service.ibbp.ovgu/"
8 version="1.0" xmlns:tns="http://service.ibbp.ovgu/"
9 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
10 <xs:element name="authentifiziereUnternehmen" type="tns:authentifiziereUnternehmen"/>
11 <xs:element name="authentifiziereUnternehmenResponse"
12 type="tns:authentifiziereUnternehmenResponse"/>
13 <xs:element name="getAnzahlAusbildungsvertraege" type="tns:getAnzahlAusbildungsvertraege"/>
14 <xs:element name="getAnzahlAusbildungsvertraegeResponse"
15 type="tns:getAnzahlAusbildungsvertraegeResponse"/>
16 <xs:element name="getLehrstellenangebote" type="tns:getLehrstellenangebote"/>
17 <xs:element name="getLehrstellenangeboteResponse"
18 type="tns:getLehrstellenangeboteResponse"/>
19 <xs:element name="identifiziereAusbildungsberuf" type="tns:identifiziereAusbildungsberuf"/>
20 <xs:element name="identifiziereAusbildungsberufResponse"
21 type="tns:identifiziereAusbildungsberufResponse"/>
22 <xs:element name="identifiziereBewerber" type="tns:identifiziereBewerber"/>
23 <xs:element name="identifiziereBewerberResponse" type="tns:identifiziereBewerberResponse"/>
24 <xs:element name="identifiziereUnbesetzteAusbildungsstellen"
25 type="tns:identifiziereUnbesetzteAusbildungsstellen"/>
26 <xs:element name="identifiziereUnbesetzteAusbildungsstellenResponse"
27 type="tns:identifiziereUnbesetzteAusbildungsstellenResponse"/>
28 <xs:element name="registriereAusbildungsstelle" type="tns:registriereAusbildungsstelle"/>
29 <xs:element name="registriereAusbildungsstelleResponse"
30 type="tns:registriereAusbildungsstelleResponse"/>
31 <xs:element name="schliesseAusbildungsvertrag" type="tns:schliesseAusbildungsvertrag"/>
32 <xs:element name="schliesseAusbildungsvertragResponse"
33 type="tns:schliesseAusbildungsvertragResponse"/>
34 <xs:complexType name="schliesseAusbildungsvertrag">
35 <xs:sequence>
36 <xs:element minOccurs="0" name="arg0" type="tns:ausbildungsstellen"/>
37 <xs:element minOccurs="0" name="arg1" type="tns:auszubildende"/>
38 </xs:sequence>
39 </xs:complexType>
40 <xs:complexType name="ausbildungsstellen">
41 <xs:sequence>
42 <xs:element minOccurs="0" name="abbruch" type="xsd:int"/>
43 <xs:element name="angeboten" type="xsd:int"/>
44 <xs:element minOccurs="0" name="ausbildungsberufe" type="tns:ausbildungsberufe"/>
45 <xs:element minOccurs="0" name="beginn" type="xsd:int"/>
46 <xs:element name="ende" type="xsd:int"/>
47 <xs:element minOccurs="0" name="fachrichtung" type="xsd:string"/>
48 <xs:element name="id" type="xsd:int"/>
49 <xs:element name="status" type="xsd:int"/>
50 </xs:sequence>
51 </xs:complexType>
52 <xs:complexType name="ausbildungsberufe">
53 <xs:sequence>
54 <xs:element minOccurs="0" name="bezeichnung" type="xsd:string"/>
55 <xs:element name="id" type="xsd:int"/>
56 </xs:sequence>
57 </xs:complexType>
58 <xs:complexType name="auszubildende">
59 <xs:sequence>
60 <xs:element name="geburtsjahr" type="xsd:int"/>
61 <xs:element name="geschlecht" type="xsd:byte"/>
62 <xs:element name="id" type="xsd:int"/>
63 </xs:sequence>
64 </xs:complexType>
65 </xs:schema>
66 </wsdl:types>
67 </wsdl:definitions>
68 </pre>
```


ServiceCollection.xml

```
49     <xs:element minOccurs="0" name="staatsangehoerigkeit" type="xs:string"/>
50   </xs:sequence>
51 </xs:complexType>
52 <xs:complexType name="schliesseAusbildungsvertragResponse">
53   <xs:sequence/>
54 </xs:complexType>
55 <xs:complexType name="getLehrstellenangebote">
56   <xs:sequence>
57     <xs:element minOccurs="0" name="arg0" type="xs:int"/>
58   </xs:sequence>
59 </xs:complexType>
60 <xs:complexType name="getLehrstellenangeboteResponse">
61   <xs:sequence>
62     <xs:element maxOccurs="unbounded" minOccurs="0" name="return"
63       type="tns:ausbildungsstellen"/>
64   </xs:sequence>
65 </xs:complexType>
66 <xs:complexType abstract="true" name="abstractCollection">
67   <xs:sequence/>
68 </xs:complexType>
69 <xs:complexType abstract="true" name="abstractList">
70   <xs:complexContent>
71     <xs:extension base="tns:abstractCollection">
72       <xs:sequence/>
73     </xs:extension>
74   </xs:complexContent>
75 </xs:complexType>
76 <xs:complexType name="arrayList">
77   <xs:complexContent>
78     <xs:extension base="tns:abstractList">
79       <xs:sequence/>
80     </xs:extension>
81   </xs:complexContent>
82 </xs:complexType>
83 <xs:complexType name="authentifiziereUnternehmen">
84   <xs:sequence>
85     <xs:element minOccurs="0" name="arg0" type="xs:string"/>
86   </xs:sequence>
87 </xs:complexType>
88 <xs:complexType name="authentifiziereUnternehmenResponse">
89   <xs:sequence>
90     <xs:element minOccurs="0" name="return" type="tns:unternehmen"/>
91   </xs:sequence>
92 </xs:complexType>
93 <xs:complexType name="unternehmen">
94   <xs:sequence>
95     <xs:element name="id" type="xs:int"/>
96     <xs:element minOccurs="0" name="Land" type="xs:string"/>
97     <xs:element minOccurs="0" name="name" type="xs:string"/>
98     <xs:element minOccurs="0" name="ort" type="xs:string"/>
99   </xs:sequence>
100 </xs:complexType>
101 <xs:complexType name="registriereAusbildungsstelle">
102   <xs:sequence>
103     <xs:element minOccurs="0" name="arg0" type="tns:ausbildungsberufe"/>
104     <xs:element minOccurs="0" name="arg1" type="tns:unternehmen"/>
105   </xs:sequence>
106 </xs:complexType>
107 <xs:complexType name="registriereAusbildungsstelleResponse">
108   <xs:sequence/>
109 </xs:complexType>
110 <xs:complexType name="identifiziereBewerber">
```

ServiceCollection.xml

```
110 <xs:sequence>
111   <xs:element minOccurs="0" name="arg0" type="tns:ausbildungsstellen"/>
112 </xs:sequence>
113 </xs:complexType>
114 <xs:complexType name="identifiziereBewerberResponse">
115   <xs:sequence>
116     <xs:element maxOccurs="unbounded" minOccurs="0" name="return"
117       type="tns:auszubildende"/>
117   </xs:sequence>
118 </xs:complexType>
119 <xs:complexType name="getAnzahlAusbildungsvertraege">
120   <xs:sequence>
121     <xs:element minOccurs="0" name="arg0" type="xs:int"/>
122   </xs:sequence>
123 </xs:complexType>
124 <xs:complexType name="getAnzahlAusbildungsvertraegeResponse">
125   <xs:sequence>
126     <xs:element maxOccurs="unbounded" minOccurs="0" name="return"
127       type="tns:ausbildungsstellen"/>
127   </xs:sequence>
128 </xs:complexType>
129 <xs:complexType name="identifiziereUnbesetzteAusbildungsstellen">
130   <xs:sequence>
131     <xs:element minOccurs="0" name="arg0" type="tns:unternehmen"/>
132   </xs:sequence>
133 </xs:complexType>
134 <xs:complexType name="identifiziereUnbesetzteAusbildungsstellenResponse">
135   <xs:sequence>
136     <xs:element maxOccurs="unbounded" minOccurs="0" name="return"
137       type="tns:ausbildungsstellen"/>
137   </xs:sequence>
138 </xs:complexType>
139 <xs:complexType name="identifiziereAusbildungsberuf">
140   <xs:sequence>
141     <xs:element minOccurs="0" name="arg0" type="xs:string"/>
142   </xs:sequence>
143 </xs:complexType>
144 <xs:complexType name="identifiziereAusbildungsberufResponse">
145   <xs:sequence>
146     <xs:element minOccurs="0" name="return" type="tns:ausbildungsberufe"/>
147   </xs:sequence>
148 </xs:complexType>
149 </xs:schema>
150 </wsdl:types>
151 <wsdl:message name="getAnzahlAusbildungsvertraege">
152   <wsdl:part element="tns:getAnzahlAusbildungsvertraege" name="parameters">
153   </wsdl:part>
154 </wsdl:message>
155 <wsdl:message name="getLehrstellenangeboteResponse">
156   <wsdl:part element="tns:getLehrstellenangeboteResponse" name="parameters">
157   </wsdl:part>
158 </wsdl:message>
159 <wsdl:message name="identifiziereUnbesetzteAusbildungsstellen">
160   <wsdl:part element="tns:identifiziereUnbesetzteAusbildungsstellen" name="parameters">
161   </wsdl:part>
162 </wsdl:message>
163 <wsdl:message name="schliesseAusbildungsvertrag">
164   <wsdl:part element="tns:schliesseAusbildungsvertrag" name="parameters">
165   </wsdl:part>
166 </wsdl:message>
167 <wsdl:message name="registriereAusbildungsstelleResponse">
168   <wsdl:part element="tns:registriereAusbildungsstelleResponse" name="parameters">
```

ServiceCollection.xml

```
169     </wsdl:part>
170 </wsdl:message>
171 <wsdl:message name="getAnzahlAusbildungsvertraegeResponse">
172   <wsdl:part element="tns:getAnzahlAusbildungsvertraegeResponse" name="parameters">
173     </wsdl:part>
174 </wsdl:message>
175 <wsdl:message name="identifiziereAusbildungsberufResponse">
176   <wsdl:part element="tns:identifiziereAusbildungsberufResponse" name="parameters">
177     </wsdl:part>
178 </wsdl:message>
179 <wsdl:message name="identifiziereAusbildungsberuf">
180   <wsdl:part element="tns:identifiziereAusbildungsberuf" name="parameters">
181     </wsdl:part>
182 </wsdl:message>
183 <wsdl:message name="identifiziereBewerber">
184   <wsdl:part element="tns:identifiziereBewerber" name="parameters">
185     </wsdl:part>
186 </wsdl:message>
187 <wsdl:message name="identifiziereUnbesetzteAusbildungsstellenResponse">
188   <wsdl:part element="tns:identifiziereUnbesetzteAusbildungsstellenResponse"
189     name="parameters">
190     </wsdl:part>
191 </wsdl:message>
192 <wsdl:message name="registriereAusbildungsstelle">
193   <wsdl:part element="tns:registriereAusbildungsstelle" name="parameters">
194     </wsdl:part>
195 </wsdl:message>
196 <wsdl:message name="identifiziereBewerberResponse">
197   <wsdl:part element="tns:identifiziereBewerberResponse" name="parameters">
198     </wsdl:part>
199 </wsdl:message>
200 <wsdl:message name="schliesseAusbildungsvertragResponse">
201   <wsdl:part element="tns:schliesseAusbildungsvertragResponse" name="parameters">
202     </wsdl:part>
203 </wsdl:message>
204 <wsdl:message name="authentifiziereUnternehmen">
205   <wsdl:part element="tns:authentifiziereUnternehmen" name="parameters">
206     </wsdl:part>
207 </wsdl:message>
208 <wsdl:message name="getLehrstellenangebote">
209   <wsdl:part element="tns:getLehrstellenangebote" name="parameters">
210     </wsdl:part>
211 </wsdl:message>
212 <wsdl:message name="authentifiziereUnternehmenResponse">
213   <wsdl:part element="tns:authentifiziereUnternehmenResponse" name="parameters">
214     </wsdl:part>
215 </wsdl:message>
216 <wsdl:portType name="ServiceCollection">
217   <wsdl:operation name="schliesseAusbildungsvertrag">
218     <wsdl:input message="tns:schliesseAusbildungsvertrag"
219       name="schliesseAusbildungsvertrag">
220     </wsdl:input>
221     <wsdl:output message="tns:schliesseAusbildungsvertragResponse"
222       name="schliesseAusbildungsvertragResponse">
223     </wsdl:output>
224 </wsdl:operation>
225 <wsdl:operation name="getLehrstellenangebote">
226   <wsdl:input message="tns:getLehrstellenangebote" name="getLehrstellenangebote">
227     </wsdl:input>
228   <wsdl:output message="tns:getLehrstellenangeboteResponse"
229     name="getLehrstellenangeboteResponse">
230   </wsdl:output>
231 </wsdl:operation>
232 </wsdl:portType>
233 </wsdl:binding>
234 </wsdl:service>
235 </wsdl:definitions>
236 </wsdl:serviceDefinition>
```

ServiceCollection.xml

```
227 </wsdl:operation>
228 <wsdl:operation name="authentifiziereUnternehmen">
229 <wsdl:input message="tns:authentifiziereUnternehmen"
name="authentifiziereUnternehmen">
230 </wsdl:input>
231 <wsdl:output message="tns:authentifiziereUnternehmenResponse"
name="authentifiziereUnternehmenResponse">
232 </wsdl:output>
233 </wsdl:operation>
234 <wsdl:operation name="registriereAusbildungsstelle">
235 <wsdl:input message="tns:registriereAusbildungsstelle"
name="registriereAusbildungsstelle">
236 </wsdl:input>
237 <wsdl:output message="tns:registriereAusbildungsstelleResponse"
name="registriereAusbildungsstelleResponse">
238 </wsdl:output>
239 </wsdl:operation>
240 <wsdl:operation name="identifiziereBewerber">
241 <wsdl:input message="tns:identifiziereBewerber" name="identifiziereBewerber">
242 </wsdl:input>
243 <wsdl:output message="tns:identifiziereBewerberResponse"
name="identifiziereBewerberResponse">
244 </wsdl:output>
245 </wsdl:operation>
246 <wsdl:operation name="getAnzahlAusbildungsvertraege">
247 <wsdl:input message="tns:getAnzahlAusbildungsvertraege"
name="getAnzahlAusbildungsvertraege">
248 </wsdl:input>
249 <wsdl:output message="tns:getAnzahlAusbildungsvertraegeResponse"
name="getAnzahlAusbildungsvertraegeResponse">
250 </wsdl:output>
251 </wsdl:operation>
252 <wsdl:operation name="identifiziereUnbesetzteAusbildungsstellen">
253 <wsdl:input message="tns:identifiziereUnbesetzteAusbildungsstellen"
name="identifiziereUnbesetzteAusbildungsstellen">
254 </wsdl:input>
255 <wsdl:output message="tns:identifiziereUnbesetzteAusbildungsstellenResponse"
name="identifiziereUnbesetzteAusbildungsstellenResponse">
256 </wsdl:output>
257 </wsdl:operation>
258 <wsdl:operation name="identifiziereAusbildungsberuf">
259 <wsdl:input message="tns:identifiziereAusbildungsberuf"
name="identifiziereAusbildungsberuf">
260 </wsdl:input>
261 <wsdl:output message="tns:identifiziereAusbildungsberufResponse"
name="identifiziereAusbildungsberufResponse">
262 </wsdl:output>
263 </wsdl:operation>
264 </wsdl:portType>
265 <wsdl:binding name="ServiceCollectionServiceSoapBinding" type="tns:ServiceCollection">
266 <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
267 <wsdl:operation name="schliesseAusbildungsvertrag">
268 <soap:operation soapAction="" style="document"/>
269 <wsdl:input name="schliesseAusbildungsvertrag">
270 <soap:body use="literal"/>
271 </wsdl:input>
272 <wsdl:output name="schliesseAusbildungsvertragResponse">
273 <soap:body use="literal"/>
274 </wsdl:output>
275 </wsdl:operation>
276 <wsdl:operation name="authentifiziereUnternehmen">
277 <soap:operation soapAction="" style="document"/>
```

ServiceCollection.xml

```
278 <wsdl:input name="authentifiziereUnternehmen">
279 <soap:body use="literal"/>
280 </wsdl:input>
281 <wsdl:output name="authentifiziereUnternehmenResponse">
282 <soap:body use="literal"/>
283 </wsdl:output>
284 </wsdl:operation>
285 <wsdl:operation name="getLehrstellenangebote">
286 <soap:operation soapAction="" style="document"/>
287 <wsdl:input name="getLehrstellenangebote">
288 <soap:body use="literal"/>
289 </wsdl:input>
290 <wsdl:output name="getLehrstellenangeboteResponse">
291 <soap:body use="literal"/>
292 </wsdl:output>
293 </wsdl:operation>
294 <wsdl:operation name="identifiziereBewerber">
295 <soap:operation soapAction="" style="document"/>
296 <wsdl:input name="identifiziereBewerber">
297 <soap:body use="literal"/>
298 </wsdl:input>
299 <wsdl:output name="identifiziereBewerberResponse">
300 <soap:body use="literal"/>
301 </wsdl:output>
302 </wsdl:operation>
303 <wsdl:operation name="registriereAusbildungsstelle">
304 <soap:operation soapAction="" style="document"/>
305 <wsdl:input name="registriereAusbildungsstelle">
306 <soap:body use="literal"/>
307 </wsdl:input>
308 <wsdl:output name="registriereAusbildungsstelleResponse">
309 <soap:body use="literal"/>
310 </wsdl:output>
311 </wsdl:operation>
312 <wsdl:operation name="getAnzahlAusbildungsvertraege">
313 <soap:operation soapAction="" style="document"/>
314 <wsdl:input name="getAnzahlAusbildungsvertraege">
315 <soap:body use="literal"/>
316 </wsdl:input>
317 <wsdl:output name="getAnzahlAusbildungsvertraegeResponse">
318 <soap:body use="literal"/>
319 </wsdl:output>
320 </wsdl:operation>
321 <wsdl:operation name="identifiziereUnbesetzteAusbildungsstellen">
322 <soap:operation soapAction="" style="document"/>
323 <wsdl:input name="identifiziereUnbesetzteAusbildungsstellen">
324 <soap:body use="literal"/>
325 </wsdl:input>
326 <wsdl:output name="identifiziereUnbesetzteAusbildungsstellenResponse">
327 <soap:body use="literal"/>
328 </wsdl:output>
329 </wsdl:operation>
330 <wsdl:operation name="identifiziereAusbildungsberuf">
331 <soap:operation soapAction="" style="document"/>
332 <wsdl:input name="identifiziereAusbildungsberuf">
333 <soap:body use="literal"/>
334 </wsdl:input>
335 <wsdl:output name="identifiziereAusbildungsberufResponse">
336 <soap:body use="literal"/>
337 </wsdl:output>
338 </wsdl:operation>
339 </wsdl:binding>
```

ServiceCollection.xml

```
340 <wsdl:service name="ServiceCollectionService">
341   <wsdl:port binding="tns:ServiceCollectionServiceSoapBinding"
name="ServiceCollectionPort">
342     <soap:address location="http://localhost:8080/RES/ServiceCollection"/>
343   </wsdl:port>
344 </wsdl:service>
345 </wsdl:definitions>
```