Bundesamt für Wirtschaft
und Ausfuhrkontrolle (BAFA)
Frankfurter Straße 29-35
65760 Eschborn

Dokumentation

zum Betrieb der
Webservice-Architektur
der Webanwendung ELAN

Version 0.2

31.10.2005

netconsult GmbH

vorgelegt durch:
Patrik Geisel
patrik.geisel@luka.de
+49.69.48 00 05-60

Versionsführung

Version	Projektphase	Datum	Autor	Status
0.1	Draft	21.10.2005	hartmut.ludwig@luka.de	in Bearbeitung
0.2	Kundenvorlage	31.10.2005	patrik.geisel@luka.de	abgeschlossen

Weiterführende Dokumente

Version	Speicherort	Beschreibung	Datum

Inhalt

1. Einleitung

Das Bundesamt für Wirtschaft und Ausfuhrkontrolle (BAFA) hat die LUKA netconsult GmbH (LUKA) mit der Erstellung einer Webservice-Architektur beauftragt, die es ermöglichen soll Antragsdaten für die im Rahmen der ELAN-Anwendung (Elektronische Antragstellung) bereitgestellten Onlineanträge entgegenzunehmen und zu verarbeiten.

Dieses Dokument stellt die hierbei verwendete technische Lösung dar und soll den Nutzern der Schnittstelle als Leitfaden zur der Erstellung eines entsprechenden Webservice-Clients und bei der Beantwortung häufig auftretender Fragen helfen.

Es wird um Verständnis gebeten, dass im Sinne der Lesbarkeit dieses technischen Dokuments auf das explizite Ausformulieren der männlichen und weiblichen Formen im Text verzichtet wird.

Verwendete Abkürzungen

Es werden die folgenden Abkürzungen verwendet:

DTD	Document Type Definition (ist eine Notationsart zur Festlegung der Struktur eines XML-Dokuments.).
HTTP	Hypertext Transfer Protocol (ein zustandsloses Datenaustausch-Protokoll zur Übertragung von Daten wie z.B. Webseiten im World Wide Web).
JS	JavaScript
MySQL	Relationales Open Source Datenbanksystem MySQL
MIME	Multipurpose Internet Mail Extensions / Multimedia Internet Message Extensions. (Codierstandard, um Nicht-Text-Dateien wie Bilder, Multimedia- oder PDF-Dokumente so zu codieren, dass sie auch in textbasierten Übertragungssystemen wie z.B. E-Mail übermittelt werden können).
PDF	Portable Document Format (Standard für formatierte Dokumente im Web)
PEAR	PHP Extension and Application Repository (eine Bibliothek von Modulen und Erweiterungen für die Skriptsprache PHP)
PHP	PHP Hypertext Preprocessor
RPC	Remote Procedure Call (ein Netzwerkprotokoll mit dessen Hilfe Funktionsaufrufe auf entfernten Rechnern durchgeführt werden können.)
SOAP	Simple Object Access Protocol (ein Protokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und entfernte Funktionsaufrufe durchgeführt werden können).
URL	Uniform Ressource Locator (beschreibt den eindeutigen Speicherort eines Objektes im Internet).

XML	Extensible Markup Language (ein Standard zur Definition von Auszeichnungssprachen)

2. Überblick über ELAN

2.1. Manuelle Erfassung von Anträgen

Das Bundesamt für Wirtschaft und Ausfuhrkontrolle (BAFA) hat im Rahmen der Initiative Bund Online 2005 seinen Service dahingehend erweitert, dass verschiedene Anträge künftig mit Hilfe der elektronischen Antragstellung (ELAN) auf der Website der BAFA erfasst werden können.

Zielgruppe für diesen Service sind sowohl private Anwender, als auch Unternehmen, für die auf diese Art eine Alternative zur teuren und fehlerbehafteten Prozedur des Beschaffens und Ausfüllens von Antragsformularen auf Papier geschaffen werden soll.

Über die Webanwendung ELAN können Benutzer diese Formulare nun online aufrufen und ausfüllen. Sie werden dabei mit verschiedenen Ausfüllhilfen unterstützt.

Die Eingabefelder werden von dem entsprechenden Fachbereich der BAFA mit einem Satz von Validierungs- und Plausibilitätsprüfungen belegt werden, die zusätzlich sicherstellen, dass die eingegebenen Daten formal und inhaltlich wesentliche Kriterien für die Weiterverarbeitung beim BAFA erfüllen. So wird die Verarbeitung dadurch beschleunigt, dass keine Anträge mehr aufgrund formaler Fehler zurückgewiesen werden müssen.

Sobald sich ein Benutzer bei dem System registriert hat, werden seine Eingaben bei jedem Schritt automatisch gespeichert. Er kann seine Arbeit also jederzeit beliebig lange unterbrechen. Nach der erneuten Anmeldung stehen ihm alle bislang eingegebenen Antragsdaten auf Wunsch wieder zur Verfügung.

Sobald die Daten eines Antrags vollständig erfasst wurden, kann der ausgefüllte Antrag in Form eines PDF-Dokumentes abgerufen werden. Durch diese PDF-Generierung ist sichergestellt, dass die jeweils aktuellen Formularvordrucke verwendet werden und sich alle Eingaben korrekt formatiert in den dafür vorgesehenen Feldern befinden.

2.2. Automatische Übermittlung von Anträgen

Als weiteren Schritt einer Strategie, die langfristig auf die papierlose Abwicklung von Anträgen zielt, stellt die BAFA nun einen Webservice zur Verfügung, durch den sich das manuelle Erfassen von Anträgen über die Weboberfläche erübrigt.

Da zahlreiche Anwender (Insbesondere aus Unternehmen) in der Regel alle für die Anträge benötigten Informationen bereits in ihren eigenen IT-Systemen verwalten, ist eine erneute manuelle Eingabe über den ELAN Web-Frontend sehr mühsam und fehleranfällig. Verschiedene Unternehmen haben hierfür auch bereits eigene Lösungen im Einsatz, die die erfassten Daten formatieren und auf einem entsprechenden Formularvordruck ausdrucken.

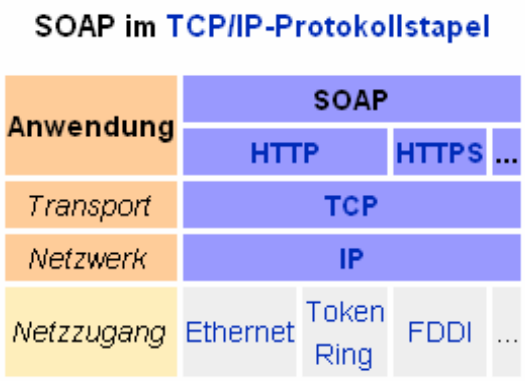
Für diese Benutzergruppe steht mit dem ELAN Webservice nun eine öffentliche, standardisierte Schnittstelle zur Verfügung, über die die Antragsdaten übermittelt werden können. Diese werden dann nach den gleichen Kriterien wie auch die Anträge über das Webformular geprüft. So können diese Anwender weiterhin direkt aus ihrem IT-System heraus Anträge stellen und profitieren dennoch von der erhöhten Sicherheit durch Prüfregeln und dem automatischen Ausfüllen der Anträge.

3. Architektur des Webservice

3.1. SOAP-Architektur

Der WebService wurde auf Grundlage des weit verbreiteten XML-Standards Simple Object Access Protocol (SOAP) http://www.w3.org/TR/soap/ implementiert. Es handelt sich dabei um ein Protokoll auf der Applikationsschicht, das entfernte Programmaufrufe über das Web in http-Requests kapselt und so den Austausch von Daten zwischen Anwendungen ermöglicht, die auf verschiedenen Betriebssystemen mit verschiedenen Technologien und Programmiersprachen implementiert wurden.

Die Verwendung des SOAP-Protokolls bietet eine Reihe von Vorteilen gegenüber den herkömmlichen Methoden (wie z.B. RPC). Diese führen häufig zu Problemen mit der Kompatibilität und Sicherheit. Außerdem blocken Firewalls und Proxy-Server in der Regel die notwendigen Ports, was den Betrieb solcher Lösungen zusätzlich erschwert oder riskant macht.



Solche Probleme sind bei SOAP ausgeschlossen, weil die Kommunikation unter Verwendung des http-Protokolls erfolgt, das von allen Browsern und Servern unterstützt wird. Außerdem können Clientanwendungen zum Aufruf des ELAN-Programms auf den unterschiedlichsten Betriebssystemen und Programmiersprachen implementiert werden.

3.2. Aufbau der SOAP-Nachrichten

Eine SOAP-Nachricht ist ein gewöhnliches XML-Dokument, welches die folgenden Elemente enthält:

- Einen obligatorisches Envelope-Element (Umschlag) welches das XML-Dokument als eine SOAP-Message ausweist.
- Ein optionales Header-Element, welches Kopfinformationen enthält.

- Ein obligatorisches Body-Element, das die Aufruf- und Antwortinformationen enthält.
- Ein optionales Fehler-Element, welches Informationen über Fehler die bei der Verarbeitung der Nachricht auftragen enthält.

Alle hier erwähnten Elemente müssen im default-Namespace des SOAP-Envelopes deklariert werden (http://www.w3.org/2001/12/soap-envelope).

Der Standard-Namespace für das SOAP-Encoding und –Datentypen ist: http://www.w3.org/2001/12/soap-encoding.

Die genaue Beschreibung einer gültigen SOAP-Nachricht des ELAN-Systems erfolgt in Form einer WSDL-Schnittstellenbeschreibung (s. Kapitel 5.1). Diese kann allen interessierten Nutzern z.B. per Link auf der BAFA-Website öffentlich zugänglich gemacht werden.

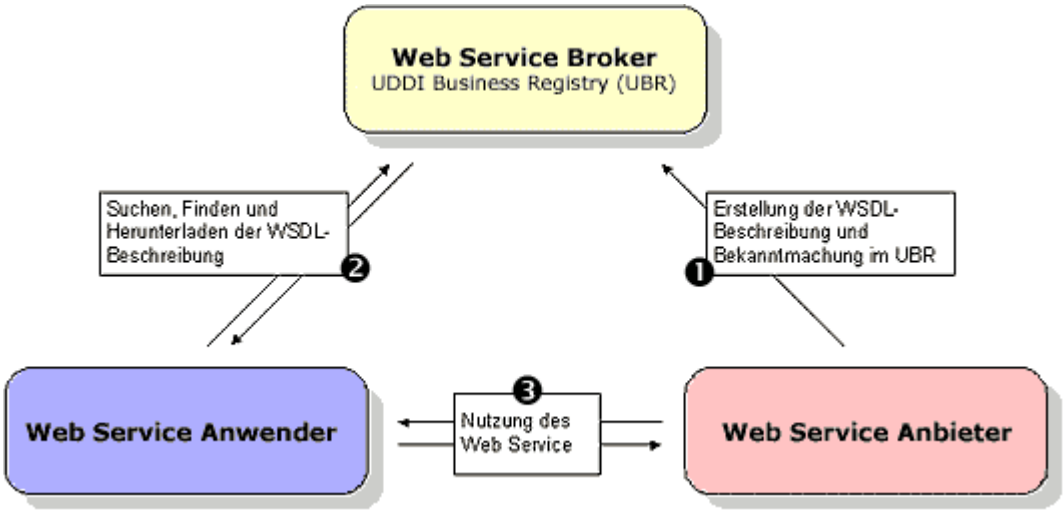
Innerhalb des Body-Bereiches der SOAP-Nachricht befindet sich die eigentliche Information des ausgefüllten Antrages. Dieser muß gegen die der DTD eines Elan-Antrages erfolgreich validierbar sein. Auch diese DTD wird über die Website der BAFA zur Verfügung gestellt werden (s. Kapitel 5.2).

3.3. Client-Server-Kommunikation

Der Webservice selbst wird als Server implementiert, der Aufrufe von einer Client-Software entgegennimmt. Das heißt, der Web Service ist nicht für menschliche Benutzer gedacht, sondern für Softwaresysteme, die automatisiert Daten austauschen und hierzu Funktionen auf entfernten Rechnern aufrufen müssen.

Damit der Anwender des Webservice weiß, welche Funktionen zur Verfügung gestellt werden und in welcher Form der Datenaustausch stattfinden soll, veröffentlicht der Anbieter (in diesem Fall die BAFA) eine Beschreibung des Dienstes in Form einer WSDL-Datei (Webservice Description Language).

Diese kann bei Bedarf auch in einem UDDI Verzeichnisdienst publiziert werden, damit Webservices unter thematischen kriterien gefunden werden können (siehe http://www.uddi.org).



Auf diesen Weg können Anwender den Webservice finden und Informationen über die bereitgestellten Komponentendienste abfragen. Nachdem eventuell weitere Protokolldetails ausgetauscht werden, findet die dynamische Anbindung des Konsumenten an den Anbieter statt. Der Konsument greift nun auf Methoden zu.

Die Grundlage hierbei bilden die folgenden XML-basierten Standards

- UDDI als Verzeichnisdienst zur Registrierung von Web Services ermöglicht das dynamische Finden des Web Services durch den Anwender.
- WSDL zur Beschreibung der unterstützen Methoden (z. B. zur Übermittlung eines Antrags) und deren Parameter (z. B. Antragsdaten) für den Programmierer des Webservice-Clients.
- SOAP beschreibt das Verfahren für die Kapselung der Aufrufe in XML und den den Transport, sowie die Übermittlung der Rückgabewerte oder eventueller Fehlerinformationen.

Erreichbar sind Web Services über eine eindeutige URI. Die verwendeten plattformunabhängigen Standards sind in der Lage, entfernte Methodenaufrufe beliebiger Plattformen zu dekodieren und einer Anwendung weiterzuleiten. Auf diese Weise entsteht eine verteilte Architektur. Die Kommunikation mit Web Services erfolgt über Nachrichten, die über mehrere Protokolle transportiert werden können. Nähere Informationen zum Thema Webservice und weiterführende Links können auf der Website der LUKA netconsult GmbH unter (http://www.luka.de/topthema/0204/index.html) abgerufen werden.

3.4. Aufrufadresse des ELAN WebService

Der Webervice wurde unter Verwendung der Programmiersprache PHP und des Implementierung des WebService mit PEAR (PHP Extension and Application Repository) implementiert. Er ist über die folgende URI aufrufbar:

https://fg01.bafa.bund.de/elan/webservice/server.php

3.5. Mitteilung von Fehlern

Die Mitteilung von Fehlern bei Verarbeitung innerhalb der Methoden des Webservice wird mit dem ‚SOAP Fault Element' realisiert. Dieses Element ist vom SOAP-Standard für Fehlerbehandlung vorgesehen. Es wird von der Client-Komponente ausgewertet, so dass der Anwender weiß, was genau bei der Verarbeitung falsch gelaufen ist und was er folglich ändern muß, um den Antrag korrigiert einsenden zu können.

3.6. Mehrsprachigkeit

Alle Fehlermeldungen des Systems werden jeweils in der Sprache ausgegeben, die auch beim Einreichen des Antrags spezifiziert wurde. Voraussetzung hierfür ist, dass die Texte für die entsprechende Sprache von der BAFA erfasst und freigegeben wurden.

Wenn dies nicht der Fall ist, weil die Sprache für den entsprechenden Formularworkflow noch nicht aktiv geschaltet wurde, dann wird die Annahme des Antrags unter dieser Sprache verweigert. Sie können eventuell den gleichen Antrag in einer anderen Sprache einreichen, damit er angenommen und verarbeitet wird.

4. Anwendung des Webservice

4.1. Der Webservice-Client

Zur Anwendung eines Webservice benötigt man einen Webservice-Client, der die Aufrufe in dem erwarteten Format durchführt. Der Webservice-Client wird von dem Anwender basierend auf den veröffentlichen Schnittstelleninformationen (also der WSDL-Spezifikation) erstellt. Dies kann in einer beliebigen Programmiersprache geschehen.

Die bei der Erstellung dieses Dokuments aktuelle WSDL-Spezifikation des ELAN Webservice ist in Kapitel 5.1 zu finden. Die jeweils verbindlich gültige WSDL-Beschreibung kann jederzeit online abgerufen werden. Hierzu wird an die vorgenannte Aufrufadresse des Webservice der Parameter ?WSDL angehängt:

https://fg01.bafa.bund.de/elan/webservice/server.php?WSDL

4.2. Der Webservice-Server

Der Webservice kann XML-Quellen in ein Array entsprechend dem Userinput eines Formularworkflows umwandeln, so als hätte ihn der Benutzer manuell über den Webfrontend der Anwendung eingegeben.

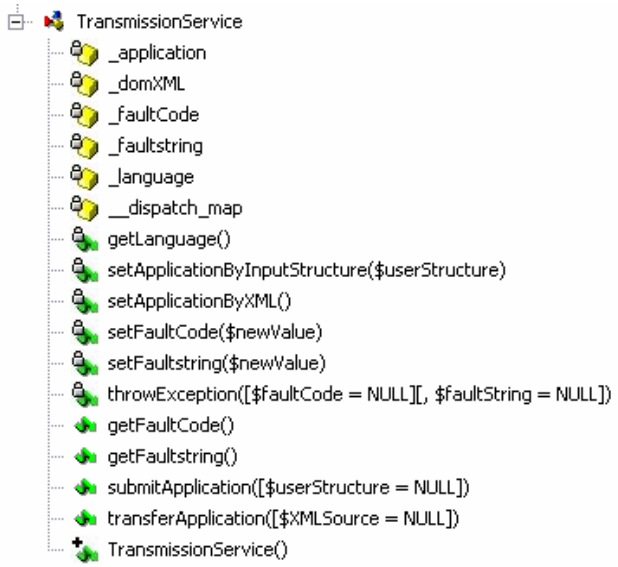
Die Validierung erfolgt ebenfalls analog dem Frontend und wurde dahingehend angepasst, dass komplette Formularworkflows in einem Durchgang abgearbeitet werden können.

Die Response-Methode gibt bei einem Fehler der Validierung ein ‚SOAP Fault Element' zurück, das eine genaue Fehlermeldung darüber beinhaltet, an welcher Stelle der Validierung (formal oder inhaltlich) der Fehler aufgetreten ist. Bei formalen Fehlern wird die Fehlermeldung des XML-Parsers übermittelt, bei inhaltlichen Fehlern wird die gleiche Fehlermeldung übermittelt, die der Frontend-User auch bei der manuellen Eingabe der fehlerhaften Daten im Webfrontend erhalten hätte.

Im Erfolgsfall enthält die Client-Message einen Link zum Download des entsprechenden Antrages.

4.3. Funktionsweise des ELAN WebService

Die Klasse, die den Webservice der ELAN-Anwendung implementiert heißt TransmissionService. Alle Felder und Methoden sind der hier abgebildeten Klassesnstruktur zu entnehmen.



Für die Benutzer des Webservice sind vor allem die beiden öffentlichen Businessmethoden **transferApplication()** und **submitApplication()** von Bedeutung.

Wie man an den Methodensignaturen erkennen kann, haben beide Methoden je einen Aufrufparameter. Bei transferApplication() ist dies einen String, der den Antrag und die notwendigen Autorisierungsdaten in Form einer XML-Struktur enthält, während submitApplication() mit einer einfachen Eingabestruktur (userStructure) aufgerufen wird, die die Userdaten zum Zwecke der Autorisierung und zur Spezifizierung des Workflows enthält. Die Beschreibung der User Structure ist dem folgenden XML-Schema zu entnehmen:

4.4. Web-Service-Methode transferApplication()

Die Verarbeitung der übermittelten Daten erfolgt in transferApplication() in 11 aufeinanderfolgenden Schritten, die aufgrund der zahlreichen Fehlerzustände, die auftreten und zu einem Abbruch der Verarbeitung führen können hier detailliert erläutert werden.

4.4.1. Analyse des Übergabeparameters

Sollte die Methode ohne den notwendigen Übergabeparameter aufgerufen werden, wird dies mit einem SOAP-Fehler quittiert.

SOAP-Fehlernummer: 001

4.4.2. Überprüfung ob die XML-Struktur wohlgeformt ist

Geprüft wird die Wohlgeformtheit des XML-Strings gemäß XML-Spezifikation http://www.w3.org/TR/2004/REC-xml11-20040204/#sec-well-formed.

Die Rückgabe eines Fehlers erfolgt entweder durch den XML-Parser der SOAP-Implementierung oder wird kein ELAN-Fehler zurückgemeldet.

SOAP-Fehlernummer: 002

4.4.3. Überprüfung ob die XML-Struktur valide ist

Geprüft wird die Validität des übermittelten XML- Strings. Hierzu muss der Antrag erfolgreich gegen die DTD des ELAN-Antrags, wie sie in Kapitel 5.2 spezifiziert und im Kopf der Antragsdatei übermittelt wird validiert werden.

Falls diese Validierung fehlschlägt, wird dies mit einem SOAP-Fehler quittiert und die weitere Verarbeitung wird abgebrochen. Die Daten werden in diesem Fall nicht an die eigentliche ELAN-Anwendung weitergeleitet. Daher hat der Anwender unter diesen Umständen keine Möglichkeit, die Daten im Webfrontend zu bearbeiten. Erst wenn die formale Validität gewährleistet ist, wird die nächste Methode getFrontendResult() aufgerufen.

SOAP-Fehlernummer: 003

4.4.4. Erstellung des Webservice-Frontends

Innerhalb der eingehenden XML-Struktur enthält einer der übermittelten Parameter den Kurznamen des Formulars. Auf diese Weise kann der Webservice-Frontend spezifizierten, für welchen Antrag die eingehenden Daten bestimmt sind.

Für diesen Kurznamen versucht die Anwendung ein Frontend-Objekt zu erzeugen, das ganz ähnlich wie der Webfrontend arbeitet, mit dem die Benutzer manuell ihre Inhalte eingeben. Hierbei können folgende Ausnahmen auftreten:

- Das angeforderte Formular existiert nicht
- Das angeforderte Formular ist nicht aktiv geschaltet
- Das angeforderte Formular ist in der geforderten Sprache nicht aktiv geschaltet.

SOAP-Fehlernummer: 010

4.4.5. Überprüfung des Service-Listeners

Nicht jedes Formular kann auch über den Webservice eingereicht werden. Voraussetzung hierfür ist, dass die Administratoren einen Webservice-Listener hierfür aufgesetzt haben. Wenn dies für den übermittelten Antrag nicht zutrifft, wird dieser abgewiesen.

SOAP-Fehlernummer: 011

4.4.6. Überprüfung der Benutzerrechte

Über den Webservice-Frontend können nur Webservices genutzt werden, die auch eine Benutzerautorisierung erfordern. Dies wird geprüft und Anträge die für ein Formular bestimmt sind, das keine Benutzerautorisierung erfordert werden abgewiesen.

Ähnlich wie beim Login über den normalen Webfrontend, überprüft der Webservicefrontend außerdem, ob der Benutzer mit den Logindaten korrekt authentisiert ist (Identifikationsnummer, Nummernkreis, Loginname und Passwort passen zusammen) und ob er autorisiert ist, das Formular aufzurufen (Prüfung ob der Benutzer in einer der Benutzergruppen Mitglied ist, die dem Formular zugewiesen wurden). Falls eine dieser Bedingungen nicht zutrifft, wird die Fehler-Beschreibung im SOAP Faultobjekt abgelegt.

SOAP-Fehlernummer: 012

4.4.7. Übername der Vorbelegungswerte

Sobald der Benutzer angemeldet ist, überprüft das System, ob Vorbelegungswerte aus dem Benutzerprofil in das Formular automatisch übernommen werden müssen (sog. dynamische Vorbelegung).

4.4.8. Übername der Eingaben in das Formular

Aus den Antragsdaten werden nun Feld für Feld diejenigen Daten in das Formular übernommen, auf ein bestehendes BOL-Feld gemappt werden können. Felder bei denen dies nicht gelungen ist, stehen im Antrag nicht zur Verfügung.

Wenn keines der übermittelten Felder auf ein bestehendes BOL-Feld gemappt werden kann, steht der gesamte Antrag nicht zur Verfügung.

Falls nur einige Felder erfolgreich übernommen wurden, wird deren Zahl angezeigt. Falls eine oder mehrere Ausnahmebedingungen eintritt, bricht die Verarbeitung in diesem Bereich der Anwendung nicht ab, sondern protokolliert lediglich im Fault-Element die vom Frontend gemeldeten Fehlermeldungen. Konkret gibt es folgende Ausnahmebedingungen:

- Ein übermitteltes Element wurde nicht im Antrag gefunden. Der Wert wird folglich nicht zugewiesen.
- Es wurde versucht einen Wert in ein Element zu schreiben, das schreibgeschützt oder inaktiviert ist. Auch in diesem Fall wird der Wert nicht zugewiesen.
- Es wird versucht einen Wert zuzuweisen, der nicht auf den Wertebereich des Elements passt.
    - Wenn das Element zum Beispiel eine Selectbox ist, sind natürlich nur die Einträge statthaft, die auch sonst bei der Auswahl aus der Selectbox in dem Feld möglich gewesen wären. Gleiches gilt für Gruppen von Radiobuttons oder Checkboxen.
    - Falls ein Eingabefeld eine Längenbeschränkung hat (maxlengh), so darf die Länge des übermittelten Strings diese Beschränkung nicht verletzen.
    - Eine Checkbox kann nur die Werte „on“, „0“ oder „1“ haben.
- Es wird versucht ein Keytag zu übermitteln, in dem sich kein Inhalt befindet
- Es wird eine leere oder falsche Wiederholungsblockreferenz übergeben. (Wiederholungsblockreferenzen enthalten einen Punkt, der den Namen des Wiederholungsblocks von dem Namen des Elements trennt, das im Wiederholungsblock liegt, sowie einer angehängten Wiederholugszahl, die spezifiziert, zu welcher Iteration die übermittelten Elementdaten gehören.
- Es wird eine ungültig hohe Iterationszahl für Wiederholungsblöcke übermittelt.
- Es wird ein ungültiger Wiederholungsblockname übermittelt.

SOAP-Fehlernummer: 014

4.4.9. Speicherung der übernommenen Eingaben

Die Eingaben werden (genauso wie bei der manuellen Eingabe) in einer Datenbank verschlüsselt gespeichert. So hat der Anwender im Fehlerfall (oder gegebenenfalls auch zur nachträglichen inhaltlichen Bearbeitung) die Möglichkeit, seinen eingesendeten Antrag im Webfrontend zu bearbeiten.

Grundsätzlich kann immer nur der zuletzt über den Webservice-Client abgeschickte Antrag für die Bearbeitung im Webfrontend geöffnet werden. Ein erneutes Verschicken des Antrages führt dazu, dass der letzte Antrag überschrieben wird.

4.4.10. Prüfung der übernommenen Antragsdaten

Im Anschluss an die Zuweisung der Werte werden sämtliche Prüfregeln genauso aufgerufen, wie dies bei der manuellen Eingabe auch passieren würde.

Welche Prüfregeln dies sind, richtet sich nach den aktuell von den BAFA-Mitarbeitern konfigurierten Vorgaben und Validierungsregeln.

Falls hierbei ein inhaltlicher Validierungsfehler zurückliefert wird, bekommt der Client über das ‚SOAP Fault Element' die gleiche Information über die Frontend-Fehler, angezeigt wie der Benutzer der Webanwendung. Diese wird lediglich um den Variablennamen ergänzt, auf den sich die verletzte Prüfregel bezieht, denn eine farbliche Hervorhebung der betroffenen Felder (wie beim Webfrontend) steht in diesem Fall ja nicht zur Verfügung.

SOAP-Fehlernummer: 015

4.4.11. Mitteilung des Ergebnisses

Falls getFrontendResult() eine erfolgreiche Prüfung zurückliefert, wird ein String zurückgeliefert, der bei der Einstellung der Deutschen Locale den Rückgabewert „Alle Antragsdaten wurden erfolgreich übernommen“ enthält.

4.5. Web-Service-Methode submitApplication()

Bei erfolgreichem Durchlauf der Methode transferApplication() kann der Antrag durch den Aufruf von submitApplication() an das BOL-System eingereicht werden. Die Methode liefert im Erfolgsfall eine Link-Referenz zu dem resultierenden PDF-Dokument zurück. Die folgenden Schritte werden hierbei abgearbeitet:

4.5.1. Analyse des Übergabeparameters

Der Analyseparameter ist eine Eingabestruktur (UserInputStructure) gemäß der XSD-Spezifikation UserStructure.xsd, die auf der Startseite des Webservice verlinkt ist.

SOAP-Fehlernummer: 006

4.5.2. Überprüfung, ob das Element uniqueWorkflowName gesetzt ist

Mindestanforderung an einen Aufruf ist die Übermittlung des uniqueWorkflowName Elements in der UserStructure. Falls dieses Element nicht gesetzt ist, wird ein SOAP Fehler geworfen.

SOAP-Fehlernummer: 007

4.5.3. Überprüfung, ob das Element language gesetzt ist

Eine weitere Mindestanforderung an den Aufruf ist die Übermittlung des Sprachparameters language. Falls dieses Element nicht gesetzt ist, wird ein SOAP Fehler geworfen.

SOAP-Fehlernummer: 008

4.5.4. Zuweisung des Übergabeparameters

Falls weitere Fehler in der Eingabestruktur auftreten, wird dies ebenfalls mit einem SOAP Fehler quittiert.

SOAP-Fehlernummer: 009

4.5.5. Erstellung des Webservice-Frontends

Der Service-Frontend wird instanziert. Alle diesbezüglichen Ausnahmen sind bereits in Kapitel 4.4.4 näher erläutert

SOAP-Fehlernummer: 010

4.5.6. Überprüfung des Service-Listeners

Der Service-Listener wird überprüft. Alle diesbezüglichen Ausnahmen sind bereits in Kapitel 4.4.5 näher erläutert.

SOAP-Fehlernummer: 011

4.5.7. Überprüfung der Benutzerrechte

Die Nutzerrechte werden überprüft. Alle diesbezüglichen Ausnahmen sind bereits in Kapitel 4.4.6 näher erläutert.

SOAP-Fehlernummer: 012

4.5.8. Laden der zuvor Antragsdaten aus der Datenbank

Alle Daten, die zuvor mit transferApplication() übermittelt wurden werden ausgelesen. Falls keine derartigen Benutzerdaten für den aktuell eingeloggten Benutzer und das aktuelle Formular existieren, wird dies mit einem SOAP Fehler quittiert.

SOAP-Fehlernummer: 013

4.5.9. Überprüfung der Antragsdaten

Es erfolgt eine Prüfung der Antragsdaten wie in 4.4.10 beschrieben. Fehler werden mit der SOAP-Fehlernummer: 015 quittiert.

4.5.10. Auslieferung der PDF-URI

Falls alle Daten sauber dem PDF-Antrag zugewiesen werden konnten, erfolgt die Rückgabe der URI, die zum ausgefüllten PDF-Antrag führt. Fehler, die hierbei auftreten könnten bestehen zum Beispiel darin, dass der Nummernkreis für Antragsnummern, die vergeben werden können erschöpft ist, oder Fehler in

Workflow-Include-Dateien auftreten. Dies wird mit einem entsprechenden SOAP-Fault quittiert.

SOAP-Fehlernummer: 016

4.6. Testen des SOAP-Aufrufes

Zum Testen des SOAP-Aufrufes muss zunächst eine gültige Antragsdatei erstellt worden sein, die den Inhalt eines vollständig ausgefüllten Antrag gemäß der veröffentlichten DTD des Antrags (s. Kapitel 5.2) enthält. Beispieldateien für den Testaufruf der Anträge AG und AZG liegen auf der Index-Seite des Webservice-Verzeichnisses:

https://fg01.bafa.bund.de/elan/webservice.

Folgende Optionen stehen zum Testen zur Verfügung:

- mit Hilfe eines eigenen Client Programms das Testdaten in der entsprechenden Form aufbereitet an den Server schickt.
- viele gängige XML- oder Webservice-Erstellungstools stellen Funktionen zum Testen von SOAP-Messages zur Verfügung
  Eines der Tools, die man in diesem Zusammenhang verwenden kann ist z.B. Stylus XML: http://www.stylusstudio.com/ws_tester.html oder das Microsoft SOAP Toolkit.
- Von der oben erwähnten Index-Seite des Webservice gelangt man außerdem zu zwei Testclients, mit denen man die beiden Methodenaufrufe und die Ausgabe des Fault-Elements separat online testen kann.

5. Dokumentspezifikationen

5.1. Spezifikation des ELAN Webservice

5.1.1. WSDL-Beschreibung des ELAN Webservice

```
<?xml version="1.0"?>
<definitions name="TransmissionService" targetNamespace="urn:TransmissionService"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="urn:TransmissionService"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:ns5="https://fg01.bafa.bund.de/xsd/">
   <types xmlns="http://schemas.xmlsoap.org/wsdl/">
      <schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace=
         "https://fg01.bafa.bund.de/xsd/">
         <complexType name="UserStructure">
            <all>
               <element name="id" type="xsd:string" />
               <element name="identType" type="xsd:string" />
               <element name="identNumber" type="xsd:string" />
               <element name="password" type="xsd:string" />
               <element name="uniqueWorkflowName" type="xsd:string" />
               <element name="language" type="xsd:string" />
            </all>
         </complexType>
      </schema>
   </types>
   <message name="transferApplicationRequest">
      <part name="applicationXML" type="xsd:string" />
   </message>
   <message name="transferApplicationResponse">
      <part name="result" type="xsd:string" />
   </message>
   <message name="submitApplicationRequest">
      <part name="userStructure" type="ns5:UserStructure" />
   </message>
   <message name="submitApplicationResponse">
      <part name="result" type="xsd:string" />
   </message>
   <portType name="TransmissionServicePort">
      <operation name="transferApplication">
         <input message="tns:transferApplicationRequest" />
         <output message="tns:transferApplicationResponse" />
      </operation>
      <operation name="submitApplication">
         <input message="tns:submitApplicationRequest" />
         <output message="tns:submitApplicationResponse" />
      </operation>
   </portType>
   <binding name="TransmissionServiceBinding" type="tns:TransmissionServicePort">
      <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
      <operation name="transferApplication">
         <soap:operation soapAction=
         "http://schemas.xmlsoap.org/soap/envelope/#transmissionservice#transferApplicatio
         n" />
```

```
        <input>
           <soap:body use="encoded" namespace=
           "http://schemas.xmlsoap.org/soap/envelope/"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </input>
        <output>
           <soap:body use="encoded" namespace=
           "http://schemas.xmlsoap.org/soap/envelope/"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </output>
     </operation>
     <operation name="submitApplication">
        <soap:operation soapAction=
        "http://schemas.xmlsoap.org/soap/envelope/#transmissionservice#submitApplication"
        />
        <input>
           <soap:body use="encoded" namespace=
           "http://schemas.xmlsoap.org/soap/envelope/"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </input>
        <output>
           <soap:body use="encoded" namespace=
           "http://schemas.xmlsoap.org/soap/envelope/"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </output>
     </operation>
  </binding>
  <service name="TransmissionServiceService">
     <documentation />
     <port name="TransmissionServicePort" binding=
                  "tns:TransmissionServiceBinding">
        <soap:address location=
        "http://preview.luka.de/elan/webservice/server.php" />
     </port>
  </service>
</definitions>
```

5.1.2. Gültige SOAP-Message gemäß WSDL-Beschreibung

Wenn die Schnittstelle gemäß der oben abgebildeten WSDL-Beschreibung implementiert wurde, könnte eine gültige SOAP-Message zum Aufruf des Webservice so aussehen.

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
   soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
   <soap:Body>
      <m:TransmissionService xmlns:m="http://fg01.bafa.bund.de/elan/webservice/">
         <m:transferApplication>

         ... Antragsdaten ...

         </m:TransferApplication>
      </m:TransmissionService>
   </soap:Body>
</soap:Envelope>
```

An der Stelle, an der in dieser Message der Begriff „...Antragsdaten...“ steht würde im Falle eines realen Aufrufes die XML-Struktur mit den Daten des Benutzers stehen. Diese muss der im folgenden Kapitel spezifizierten DTD eines ELAN-Antrages entsprechen.

5.1.3. Schemadatei (XSD) zur Beschreibung der UserStructure

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!--W3C Schema erstellt mit XMLSpy v2005 rel. 3 U (http://www.altova.com)-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
   <xs:element name="UserStructure">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="id"/>
            <xs:element ref="identType"/>
            <xs:element ref="identNumber"/>
            <xs:element ref="password"/>
            <xs:element ref="uniqueWorkflowName"/>
            <xs:element ref="language"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="id" type="xs:string"/>
   <xs:element name="identNumber" type="xs:string"/>
   <xs:element name="identType" type="xs:string"/>
   <xs:element name="language" type="xs:string"/>
   <xs:element name="password" type="xs:string"/>
   <xs:element name="uniqueWorkflowName" type="xs:string"/>
</xs:schema>
```

5.2. Spezifikation der Elan-Application Dateien

5.2.1. DTD zur Validierung einer Elan-Application Datei

Übermittelte Antrags-Dateien werden gegen die hier abgebildete DTD validiert.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD erstellt mit XMLSpy v2005 rel. 3 U (http://www.altova.com)-->
<!ELEMENT application (userinput+)>
<!ATTLIST application
   uniqueWorkflowName CDATA #REQUIRED
   language CDATA #REQUIRED
>
<!ELEMENT elan-application (user, application, timestamp)>
<!ELEMENT key (#PCDATA)>
<!ELEMENT timestamp (#PCDATA)>
<!ELEMENT user EMPTY>
<!ATTLIST user
   id CDATA #REQUIRED
   identType CDATA #REQUIRED
   identNumber CDATA #REQUIRED
   password CDATA #REQUIRED
>
<!ELEMENT userinput (key, value)>
<!ELEMENT value (#PCDATA)>
```

5.2.2. Beispiel für eine gültige Antrags-Datei

Die hier abgebildete Struktur stellt eine sehr einfache, aber dennoch valide Antrags-Datei für den Formular-Workflow AZG (Verlängerung) dar:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE elan-application SYSTEM "http://fg01.bafa.bund.de/elan/webservice/elan-
application.dtd">
<elan-application>
   <user id="mm@musterdomain.de" identType="customsTariffCode"
         identNumber="1234567" password="test1!"/>
      <application uniqueWorkflowName="azg" language="de">
         <userinput>
            <key>neu-antragsart</key>
            <value>2</value>
         </userinput>
         <userinput>
            <key>inland-POSTFACH</key>
            <value></value>
         </userinput>
         <userinput>
            <key>gl-antr-bezug</key>
            <value>95/12345</value>
         </userinput>
         <userinput>
            <key>gl-antr-bezugdat</key>
            <value></value>
         </userinput>
         <userinput>
            <key>versicherung1</key>
            <value>on</value>
         </userinput>
         <userinput>
            <key>versicherung2</key>
            <value>on</value>
         </userinput>
         <userinput>
            <key>versicherung3</key>
            <value>on</value>
         </userinput>
         <userinput>
            <key>versicherung-aktuell</key>
            <value>on</value>
         </userinput>
      </application>
      <timestamp>xxxxxxxx</timestamp>
</elan-application>
```

6. Betrieb

Für den Betrieb gelten die gleichen Voraussetzungen wie im Dokument Betriebshandbuch_V1.0.doc der ELAN-Applikation vom 26.01.2005 dargelegt.

Zusätzlich werden für die Service-Architektur die folgenden Module verwendet:

6.1. Webservermodule (shared objects)

Im Webserver muss das Modul dom-xml zum Parsen von XML-Dateien aktiviert sein.

6.2. Pear-Module

Im PHP Includepfad müssen sich die folgenden Pakete befinden:

Paket	Version
SOAP	>= 0.9.1
HTTP_Request	>= 1.2.4
Net_URL	>= 1.0.12
Net_Socket	>= 1.0.2