

SAS ferngesteuert – Anwendungsintegration leicht gemacht

Jan Ferdinand Knoll
viadee Unternehmensberatung GmbH
Anton-Bruchhausen-Straße 8
48147 Münster
jan-ferdinand.knoll@viadee.de

Zusammenfassung

Mit der SAS Plattform steht eine multifunktionale Anwendungsumgebung zur Verfügung, mit der sich fast jede analytische Aufgabe erfüllen lässt. Hierbei wird dem Anwender eine hohe Flexibilität bei der Analyse von Daten geboten, da nahezu jede Datenbank und Schnittstelle genutzt werden kann.

Doch wie funktioniert die Anwendungsintegration in umgekehrter Richtung? Wie lassen sich aus einer Drittanwendung heraus analytische SAS Funktionen nutzen, ohne diese Anwendung verlassen zu müssen? Anhand eines Praxisbeispiels wird gezeigt, wie mit der SAS 9 Plattform analytische Funktionen als Webservice bereitgestellt und anschließend aus einer Drittanwendung aufgerufen werden können. Um Daten zwischen den Umgebungen auszutauschen, werden in diesem Beispiel XML Objekte verwendet.

Schlüsselwörter: Webservice, REST, SOAP

1 Motivation

Mit dem Enterprise BI Server steht eine Plattform zur Verfügung, die alle Funktionen zum Betrieb eines Data-Warehouse (DWH) liefert. Über die von SAS bereitgestellten Tools besteht die Möglichkeit, Daten in vielfältiger Weise auszuwerten. Auf Grund des Funktionsumfangs lässt sich die SAS Software außerdem für die unterschiedlichsten analytischen Aufgaben nutzen. Eine hohe Flexibilität wird hierbei durch die vielfältigen Schnittstellen ermöglicht, so dass annähernd jede Datenquelle in die Analyse einbezogen werden kann.

In manchen Situationen kann es jedoch wünschenswert sein, die Auswertungen in Drittanwendungen zu integrieren und die SAS Plattform lediglich über eine Serviceschicht aufzurufen. Aus diesem Grund wurden die Möglichkeiten der einfachen Anwendungsintegration im Rahmen eines Proof of Concept untersucht.

2 Anwendungskontext

Im Bereich der Geschäftsprozessanalyse gab es in den vergangenen Jahren viele Entwicklungen im Bereich der Standardisierung. Hierbei nimmt die Business Process Mo-

del and Notation (BPMN) eine immer wichtigere Rolle ein. BPMN legt den Fokus auf eine standardisierte Notation für Geschäftsprozesse und liefert eine eigene Semantik, mit der Geschäftsprozesse modelliert und dokumentiert werden können.

In diesem Kontext erfahren auch Workflow Engines eine immer größere Beliebtheit, da sie ein BPMN Modell nutzen können, um den Workflow zu digitalisieren und zu automatisieren. Im Rahmen der Workflow-Ausführung werden automatisch Prozessdaten erzeugt, anhand derer sich die Prozesse nachgelagert analysieren lassen.

Im einfachsten Fall bedeutet dies, dass für einen modellierten Prozess der jeweilige Prozessverlauf nach der Ausführung nachvollzogen werden kann. Die prozessorientierten Daten ermöglichen Auswertungen darüber, welche Bereiche des Prozessmodells mit welcher Häufigkeit durchlaufen wurden und welche Laufzeiten sich in den verschiedenen Teilbereichen ergeben haben. Neben dem klassischen DWH mit den dispositiven Geschäftsdaten entsteht auf diese Weise ein Process Warehouse (PWH), welches zur Analyse von Prozessdaten genutzt werden kann. Neben den Prozessdaten werden von den Workflow Engines in der Regel auch rudimentäre Geschäftsdaten protokolliert. So treten bspw. Vertragsnummern oder Kundenummern im Prozess auf. Diese fachlichen Schlüssel stellen die Brücke zwischen PWH und DWH dar.

Da es sich bewährt hat, die Prozessanalyse im Prozessmodell durchzuführen, besteht die Notwendigkeit, die erweiterten Geschäftsdaten des DWH in das Prozessmodell zu integrieren. Am konkreten Beispiel kann dies bedeuten, dass bei der Analyse des Prozessmodells neben den Prozessdaten auch Geschäftsdaten wie Kundenstrukturen, Kundensegmentierungen oder Deckungsbeiträge zu den Kunden angezeigt werden sollen, die den Teilbereich des Modells durchlaufen haben. Diese Daten stehen jedoch nur im DWH zur Verfügung. Ein Anwendungssprung in die dispositiven Tools sollte allerdings vermieden werden, um die Analyse möglichst einfach zu gestalten. Aus diesem Grund sollten die DWH Daten über einen Webservice bereitgestellt und in die Prozessanwendung integriert werden.

3 Integrationsmöglichkeiten

Um eine Integration zu erreichen, wurden zwei verschiedene Ansätze untersucht.

Zum einen wurde die Anwendungsintegration mit Hilfe der SAS Java API betrachtet. Über die Java API bestehen viele Möglichkeiten, SAS Funktionen in Java zu verwenden (vgl. [1] – SAS Java Client Developers Guide). So ist ein Zugriff auf die unterschiedlichsten Services von SAS möglich, wie SAS Metadata Server, SAS Workspace Server oder SAS Stored Process Server. Ein Nachteil dieses Ansatzes liegt jedoch darin, dass die Nutzung der API in Java implementiert werden muss, wodurch neben fundierten SAS Kenntnissen auch fundierte Java Kenntnisse notwendig werden. Speziell das Wissen über die Zusammenhänge einer SAS Enterprise BI Server Installation und die Wechselwirkungen der verschiedenen Serverkomponenten und deren Nutzung stellen eine vergleichsweise hohe Hürde dar.

Zum anderen wurde die lose Kopplung der Anwendungen über WebServices betrachtet (vgl. [2] SAS BI Webservices Developer Guide). Mit der SAS Enterprise BI Server Plattform stellt SAS bereits eine Applikation bereit, mit der WebServices erzeugt werden können. Ausgangspunkt ist ein SAS Stored Process. Dieser kapselt die Logik zur Ermittlung der Daten und kann von einem SAS Entwickler unabhängig entwickelt werden. Über Eingabeparameter kann die Funktionsweise des Stored Process beeinflusst werden. Die Ergebnisse werden als XML Ergebnis bereitgestellt.

Dieser Ansatz hat den Vorteil, dass die Integration in das Prozessmodell verhältnismäßig unabhängig von der Umsetzung der Analyseprozesse im Data Warehouse erreicht werden kann. Lediglich die Schnittstelle zwischen beiden Bereichen muss abgestimmt werden. Für die Datenübertragung werden Standardverfahren genutzt. Es ist nicht notwendig, ProxyKlassen in Java zu implementieren, die die WSDL implementieren, da SAS auch als RESTful WebServices genutzt werden kann, der die Ergebnisse wahlweise als XML oder JSON Objekt bereitstellt.

4 Umsetzung

Nachfolgend wird die Verwendung von Webservices zunächst anhand von zwei Beispielen dargestellt. Im Anschluss wird auf den in Kapitel 2 dargestellten Anwendungskontext eingegangen.

4.1 Ein einfaches *Hello World*-Beispiel

Im Eingangsbeispiel sollen mit Hilfe von SAS zwei Eingabeparameter addiert und das Ergebnis als Ausgabeparameter zurückgeliefert werden.

```
%global eingabe1;
%global eingabe2;
%global ausgabe;

data _null_;
  ergebnis = &eingabe1. + &eingabe2.;
  call symput("ausgabe", put(ergebnis,8.));
run;
```

Das zugehörige SAS Programm (vgl. oberhalb dargestelltes Programmbeispiel) kann als Stored Process in den Metadaten abgelegt werden. Die Registrierung des Stored Process erfolgt über die SAS Management Console in den Metadaten (vgl. Abbildung 1). Dabei können die Eingabe- und Ausgabeparameter definiert werden (vgl. Abbildung 2). Der auf diese Weise veröffentlichte Stored Process kann über die SAS Stored Process Webapplikation getestet werden um sicherzustellen, dass bei korrekter Eingabe die gewünschte Ausgabe produziert wird.

Als Standardanwendung stellt SAS die SASBIWS Applikation bereit. Diese stellt für jeden in den Metadaten hinterlegten Stored Process einen REST Endpunkt bereit, über

den der Stored Process aufgerufen werden kann. Eine weitergehende Konfiguration über die Registrierung des Stored Process hinaus ist nicht notwendig.

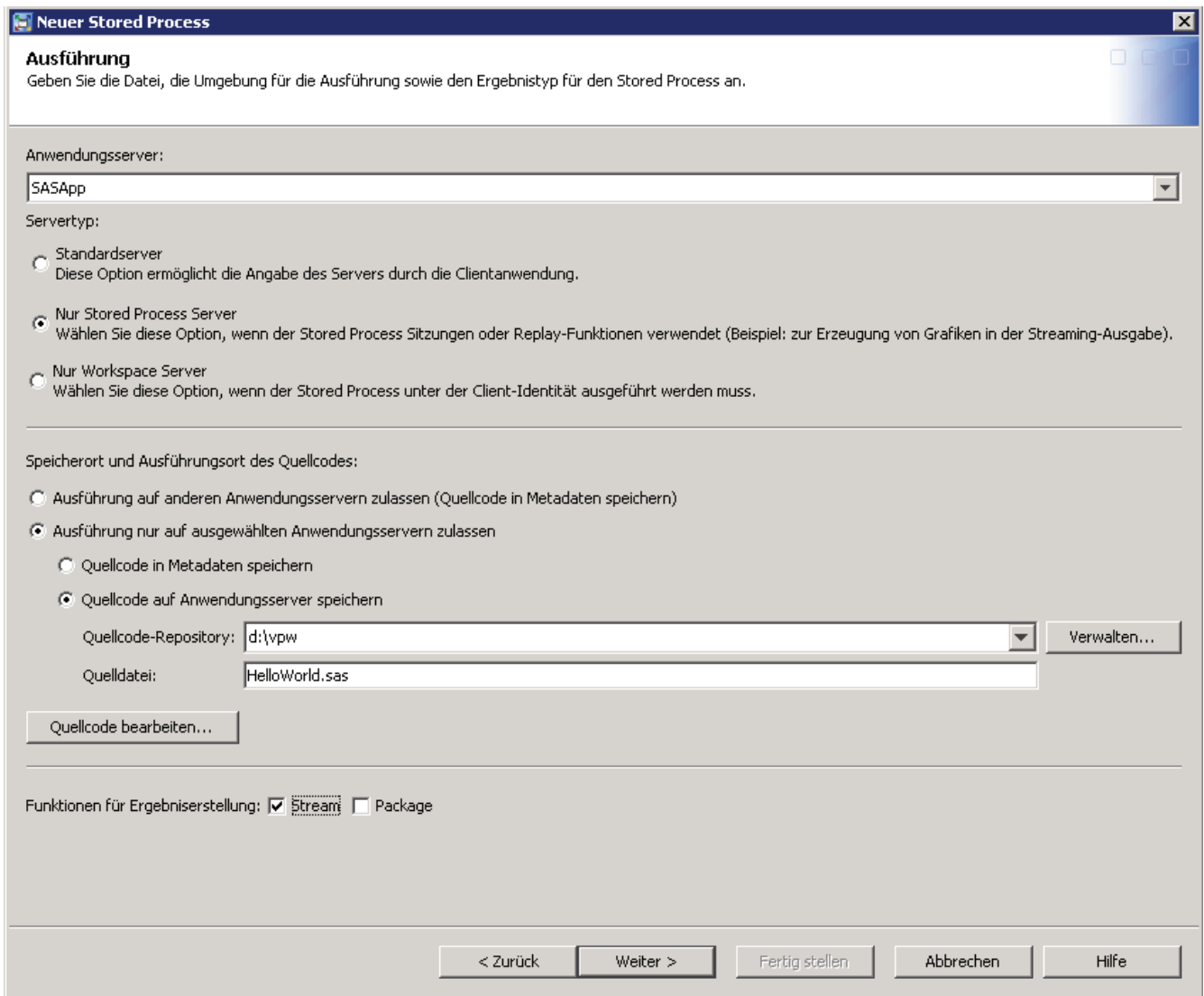


Abbildung 1: Registrierung eines Stored Process in den Metadaten

Um diesen Endpunkt zu testen, kann beispielsweise der in den Chrome Browser kostenlos integrierbare Advanced REST Client genutzt werden. In diesem Client wird die Service-Adresse angegeben. Je nach Konfiguration ist ein Authentifizierungs-Header mit einzufügen. Die Übergabeparameter werden als POST Parameter in Form eines XML Dokuments übergeben (vgl. Abbildung 3).

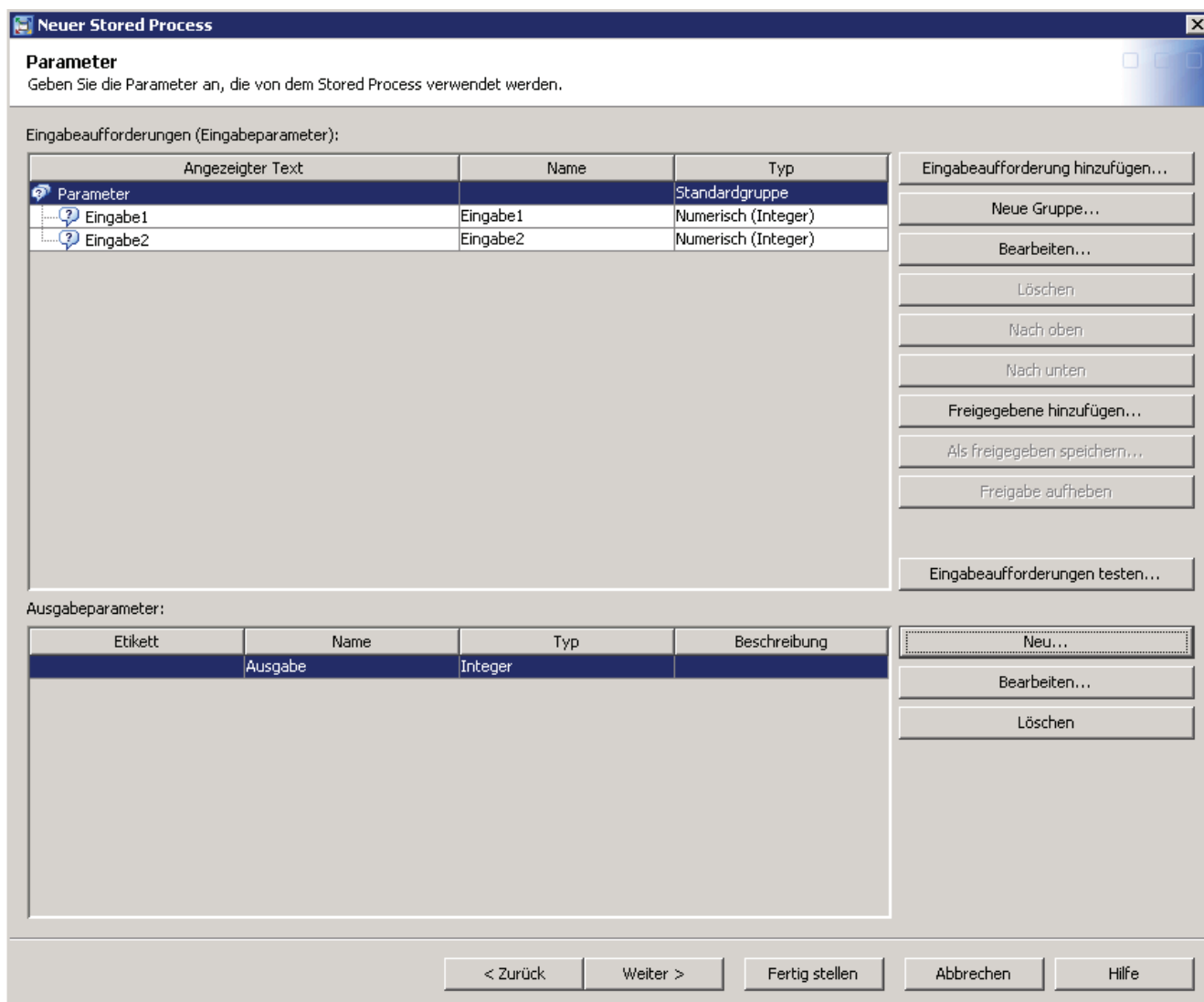


Abbildung 2: Zuweisung der Eingabe- und Ausgabeparameter

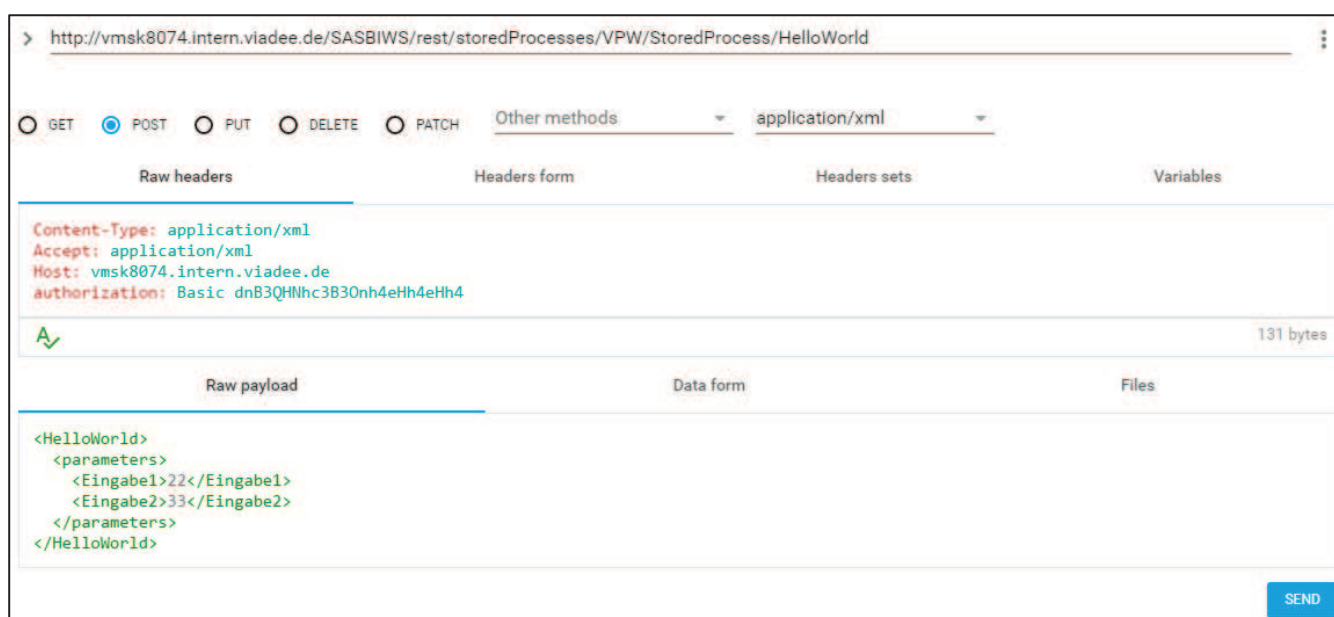


Abbildung 3: Testaufruf des Webservice

Nach dem Absenden des Requests wird das Ergebnis direkt angezeigt. Das erwartete Rechenergebnis von 55 sollte dem Ausgabeparameter übergeben werden.

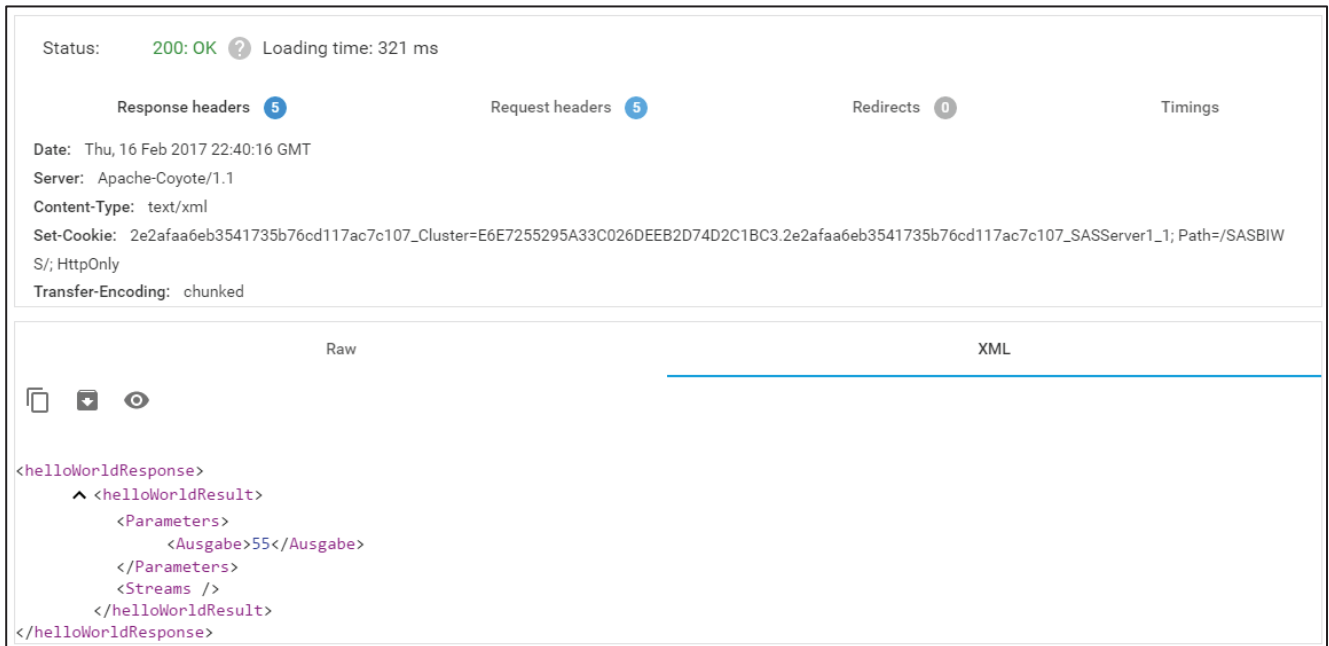


Abbildung 4: Ergebnis des WebServices

Das Ergebnis wird als XML Dokument zurückgeliefert (vgl. Abbildung 4). Alternativ kann über die URL des WebServices auch direkt auf einzelne Parameter zugegriffen werden. Erweitert man bspw. die URL wie in Abbildung 5 dargestellt, wird als Ausgabe ausschließlich der berechnete Wert ausgegeben (vgl. Abbildung 6). Ein Parsen des gelieferten XML Dokuments entfällt in diesem Fall.



Abbildung 5: URL Erweiterung

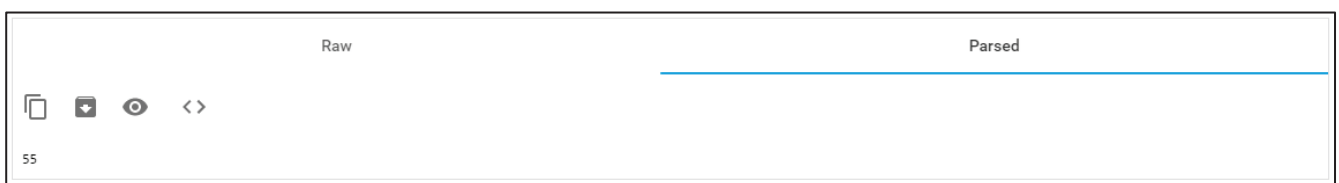


Abbildung 6: Zugriff auf Ausgabeparameter

4.2 Ein komplexeres Beispiel

Teilweise kann es jedoch gewünscht sein, nicht nur einen einzelnen Wert, sondern eine ganze Tabelle bzw. einen Ergebnisbericht über den Webservice anzufragen. Ein beispielhaftes SAS Programm für den Stored Process ist nachfolgend dargestellt.

```
%global Filter;  
libname out xml;  
data out.data;
```

```
set sashelp.class(where=(Name like "&Filter."));  
run;  
  
ods pdf file=pdf;  
proc print data=sashelp.class(where=(Name like "&Filter."));  
run;  
  
ods pdf close;
```

Im obigen Programm kann ein Filter-Parameter übergeben werden. Anschließend werden aus der SASHELP.CLASS alle Datensätze, deren Name zum Filter passt, selektiert und ausgegeben. Als Methode zur Datenübertragung wird die XML Libname Engine genutzt, die Daten automatisch in eine XML Datenstruktur überführt.

Zusätzlich sollen die Daten auch noch als Print-Ausgabe in ein PDF Dokument ausgegeben werden, welches anschließend weiterverwendet werden kann. Hierzu wird das ODS genutzt, so dass mit einem PROC PRINT die Daten in die PDF Datei ausgegeben werden.

Wichtig in diesem Zusammenhang sind die beiden Referenzen out und pdf. Diese müssen an der Stelle auf eine zugewiesene FILEREF verweisen, damit SAS auch eine zugrundeliegende Datei im Zugriff hat, in die die Daten geschrieben werden. Diese Verknüpfung der Ausgabedatei wird bei der Registrierung des WebServices in den Metadaten hergestellt. Generell unterscheidet sich die Registrierung nicht von derjenigen im vorherigen Beispiel. Lediglich die Ausgabeparameter werden nun etwas anders definiert.

Während im ersten Beispiel auf dem Reiter Parameter ein Ausgabeparameter definiert wurde, wird in diesem Beispiel auf dem Reiter Daten ein Ausgabeziel erzeugt. Dies symbolisiert den FILEREF, der von der XML Libname Engine und dem ODS genutzt wird.

Die entsprechenden FILEREFs out und pdf, die im SAS Programm für die Ausgabe genutzt werden, sind in Abbildung 7 angegeben. Der Aufruf kann analog mittels REST Client geprüft werden (vgl. Abbildung 8). Auch hierbei muss der Aufruf-Parameter Filter übergeben werden. Das Ergebnis wird als XML Dokument zurückgeliefert.

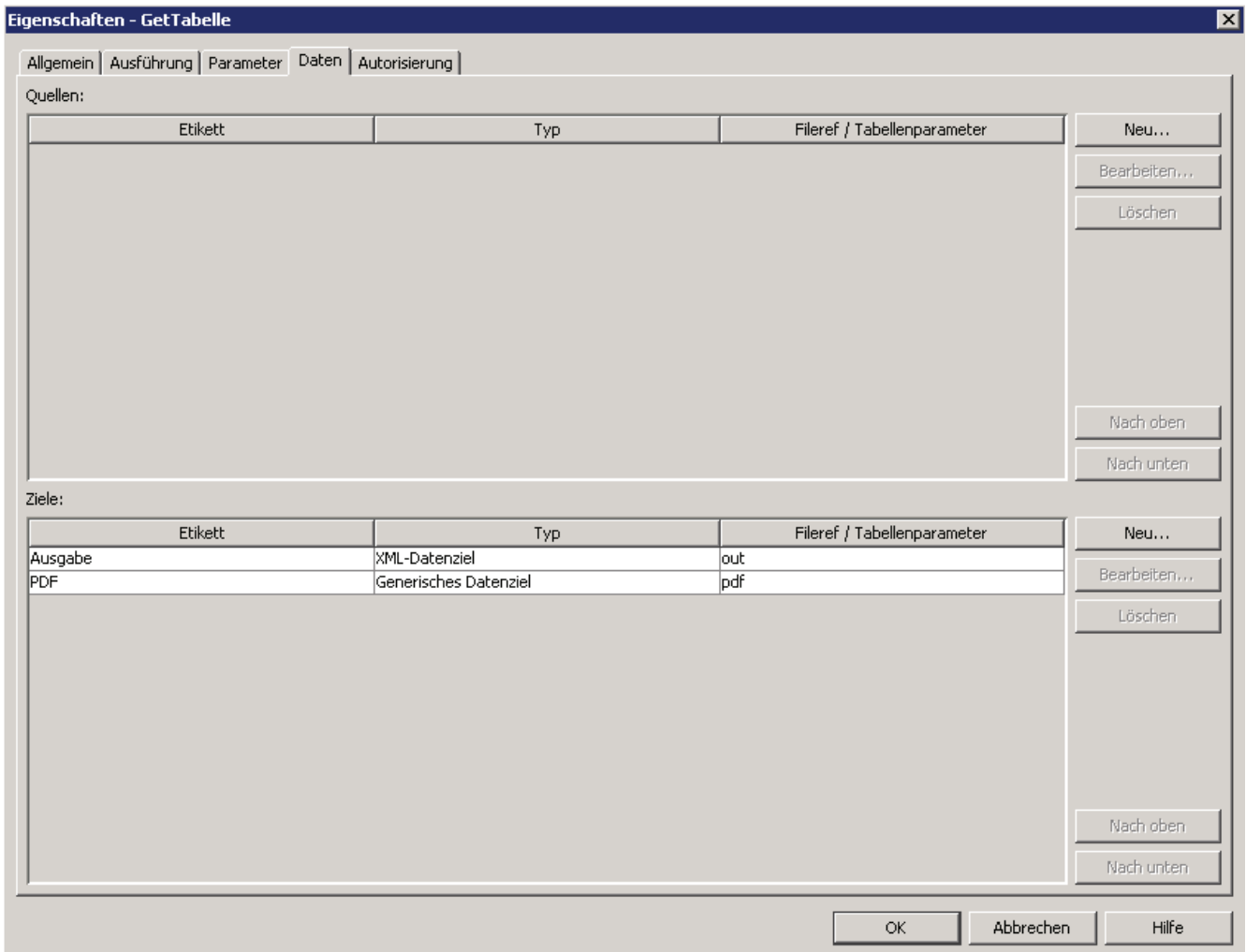


Abbildung 7: Ausgabe als Datenziel

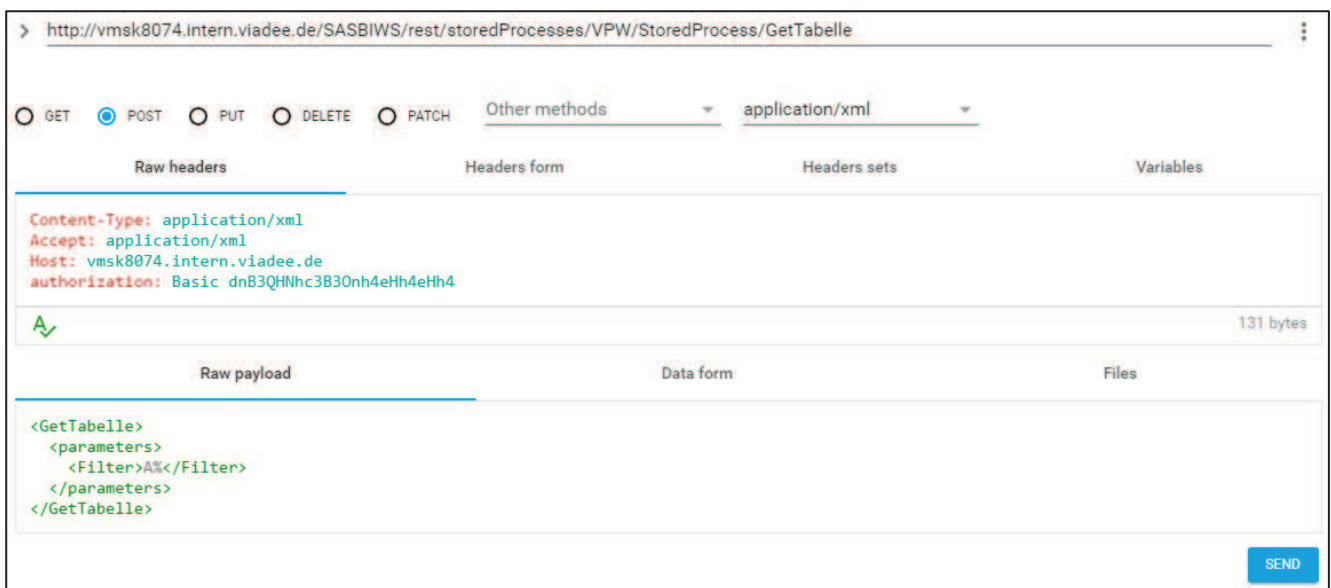


Abbildung 8: Aufruf des WebServices GetTabelle

Im vorliegenden Beispiel werden alle Datensätze gesucht, deren Name mit dem Buchstaben A beginnen.


```

Raw XML
<getTabelleResponse>
  <getTabelleResult>
    ^ <Streams>
      <out contentType="text/xml;charset=windows-1252">
        <Value>
          ^ <TABLE>
            ^ <DATA>
              <Name>Alfred</Name>
              <Sex>M</Sex>
              <Age>14</Age>
              <Height>69</Height>
              <Weight>112.5</Weight>
            </DATA>
            ^ <DATA>
              <Name>Alice</Name>
              <Sex>F</Sex>
              <Age>13</Age>
              <Height>56.5</Height>
              <Weight>84</Weight>
            </DATA>
          </TABLE>
        </Value>
      </out>
      <pdf contentType="application/pdf">
        <Value>
          JVBERi0xLjQNJbDA0AANCjcgMCMbVYmo8PC9UeXB1L0V4dEdTdGF0ZS9DQSAxLjAgL2NhIDEuMD4+ZW5kb2JqIDE1IDAgb2JqPDwvTGluZ3RoIDYyOS9Gal
        </Value>
      </pdf>
    </Streams>
  </getTabelleResult>
</getTabelleResponse>

```

Abbildung 9: Ergebnis des Webservice GetTabelle

Das Ergebnis enthält zwei Ausgabestreams (vgl. Abbildung 9). Zum einen werden die selektierten Daten im XML Format als Tabelle ausgegeben, indem für jeden Datensatz der Tabelle ein DATA Element erzeugt wird, das wiederum ein XML Tag für jede Variable enthält. Als zweiter Ausgabestream wird die PDF Datei BASE64 kodiert im XML Dokument übergeben. Dekodiert man den unter Value angegebenen String und speichert die PDF Datei, so erhält man Zugriff auf PROC PRINT des Stored Process:

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0

Abbildung 10: PDF Ausgabe des Ergebnisses

4.3 Webservice-Nutzung im Anwendungskontext

Für den in Abschnitt 2 beschriebenen Anwendungsfall konnte die Webservice basierte Integration von DWH Daten in die Prozessanwendung erfolgreich durchgeführt werden. Der in der Workflow Engine umgesetzte BPMN Prozess kann dabei direkt im Prozessmodell, in dem die Prozesskennzahlen dargestellt werden, analysiert werden. Über entsprechende Links hat der Anwender darüber hinaus die Möglichkeit, weiterführende Informationen aus dem DWH anzufragen und diese direkt im Prozessmodell anzuzeigen. Auf diese Weise werden Analysen des Geschäftsprozesses vereinfacht und ein Me-

dienbruch bei der Analyse der Daten umgangen. Ein Beispielprozess ist in Abbildung 11 dargestellt. Hier wird direkt in einem Prozessmodell angezeigt, für wie viele verschiedene Kunden die einzelnen Prozessschritte durchlaufen wurden (Prozessdaten). Zusätzlich besteht eine Filtermöglichkeit dieser Kunden nach den Kriterien *Land* oder *Geschlecht*. Bei diesen Kriterien handelt es sich um Kontextdaten, die über den Webservice aus dem DWH hinzugelesen wurden. In Abbildung 11 wurde bereits auf Kunden aus der BRD eingeschränkt. In einem nächsten Analyseschritt könnte nun nach männlichen oder weiblichen Kunden gefiltert werden.

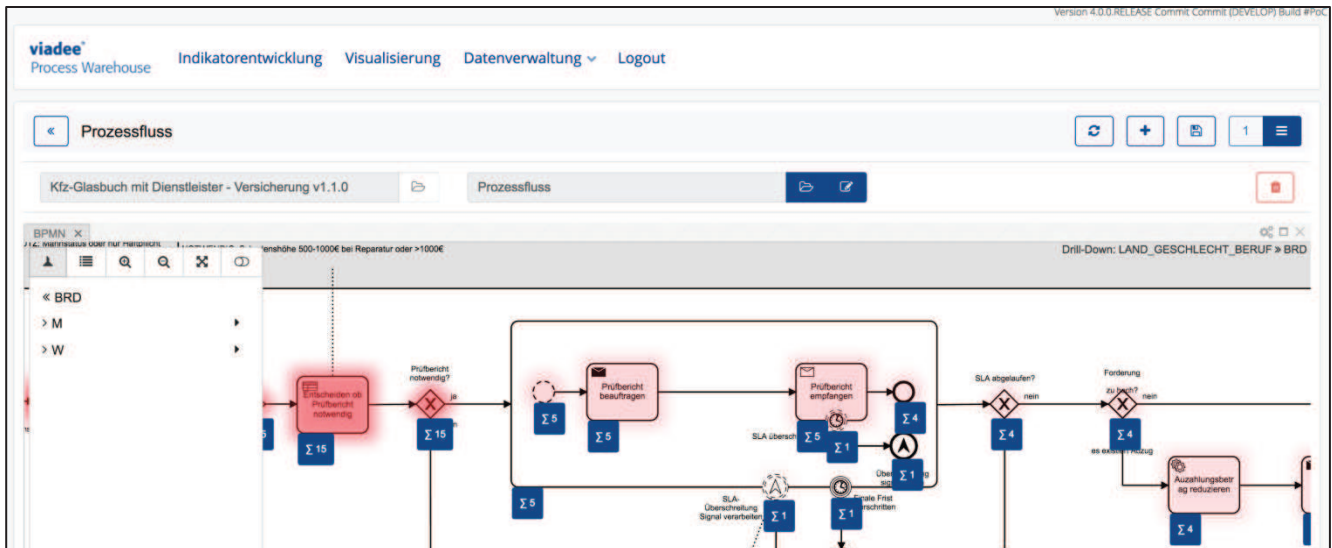


Abbildung 11: Über Webservice integrierte Filtermöglichkeit nach DWH Daten im Prozessmodell

5 Fazit

Mit Hilfe der SAS BI Webservice Applikation kann jeder in den Metadaten registrierte Stored Process als Webservice bereitgestellt werden. Hierbei besteht die Möglichkeit, auf den vollen Funktionsumfang der SAS Software zurückzugreifen. Die Ergebnisse können anschließend in verschiedenen Formaten bereitgestellt werden, z.B. als Ausgabeparameter, als Tabelle oder als Ergebnisdokument. Im Rahmen eines Proof of Concept konnte daher eine einfache Integration von DWH Daten in die Prozessmodelle einer Prozessanwendung erreicht werden.

Literatur

- [1] SAS Institute, SAS 9.4 Integration Technologies – Java Client Developers Guide, <https://support.sas.com/documentation/cdl/en/wbsvcdg/64883/PDF/default/wbsvcdg.pdf>, Abruf am 17.02.2017
- [2] SAS Institute, SAS BI Webservices Developer Guide, <https://support.sas.com/documentation/cdl/en/wbsvcdg/64883/PDF/default/wbsvcdg.pdf>, Abruf am 17.02.2017