Warehouse im Web
Web-Technologie von SAS Institute

Donald J. Henderson
überarbeitet von Dr. Markus Menke
Januar 1999

Einleitung

SAS/IntrNet und AppDev Studio sind leistungsfähige Softwarepakete zur Anbindung eines Data Warehouse an ein Intra- oder Internet. Damit ist ein entfernter Zugriff auf Unternehmensdaten möglich. Außerdem wird die Rechenleistung von SAS Software für jeden Desktop zugänglich, ganz gleich, ob die SAS Software auf dem Desktop selbst oder der zentralen Server-Plattform installiert ist.

Da bei den meisten Unternehmen unterschiedliche Technologien für die Verwaltung und Speicherung von Daten eingesetzt wird, müssen bei der Entwicklung eines Web-gestützten Systems zur Informationsverteilung zahlreiche Faktoren berücksichtigt werden. Zu diesen Faktoren, die sich auf die Wahl der geeigneten Web-Technologie auswirken, gehören unter anderem:

- Anforderungen an die Benutzerschnittstelle
- Anforderungen an den Datenzugang und die Datenanalyse
- Leistungsverhalten der Anwendung
- Kosten für Entwicklung, Anwendung und Wartung

Dieses Dokument beschreibt, wie die unterschiedlichen Komponenten von SAS/IntrNet und AppDev Studio, der Anwendungsentwicklungsumgebung zur Erstellung Java-basierter Web-Clients, innerhalb einer bestehenden DV-Infrastruktur eingesetzt werden können, um Daten in nutzbare Informationen und strategisches Wissen zu verwandeln. Hierbei wird eine Technologie eingesetzt, die für praktisch jeden zugänglich ist und mit der die meisten Anwender bereits Erfahrung haben: das World Wide Web.

Dieser Artikel enthält die folgenden Kapitel:

1. Definition von Begriffen aus der Web-Technologie
2. Komponenten von SAS/IntrNet
3. Komponenten von AppDev Studio
4. Web-Technologien und Hilsmittel zur Erstellung von Webanwendungen auf der Basis von SAS/IntrNet
5. Entwicklungsperspektiven für Web-Architekturen
6. Beziehung der SAS Web-Software zu anderer SAS Software
7. Entscheidungskriterien für die Technologieauswahl

1. Definition von Begriffen aus der Web-Technologie

Die Web-Technologie hat einen gewaltigen Einfluß auf die Informationsverteilung und den Prozeß der Anwendungsentwicklung gehabt. Sie hat das Client/Server-Modell entscheidend vereinfacht, da jetzt keine Notwendigkeit mehr besteht, unterschiedliche Anwendungen auf jedem einzelnen Desktop zu installieren. Das Web stellt die erforderliche Infrastruktur zur Verfügung, um den Einsatz von Anwendungen zu erleichtern, mit denen ein Client einzelne Dienste, wie zum Beispiel den Datenzugriff oder die Verarbeitung von Daten, auf einer anderen Maschine anfordern kann.

Webgestützte Systeme zur Informationsverteilung lassen sich nach ihrer Funktion, ihrer Arbeitsweise und den eingesetzten Technologien beschreiben. Wir beginnen daher mit einer Definition der gebräuchlichsten Begriffe, mit denen die Fähigkeiten von SAS/IntrNet und AppDev Studio beschrieben werden.

Web Publishing bezeichnet die Erstellung statischer Berichte, die zur Verfügung gestellt werden, indem sie an einem Ort gespeichert werden, der von einem Web-Server zugänglich ist. Zum Zugriff auf diese Informationen benötigt der Anwender lediglich einen Webbrowser (wie zum Beispiel Internet Explorer oder Netscape). Berichte dieser Art können auf unterschiedliche Weise erzeugt werden, zum Beispiel in Batch-Jobs. Die einzige Einschränkung besteht darin, daß das Dateiformat der Berichte vom Internet unterstützt werden muß. Dazu gehören zum Beispiel das Textformat HTML oder die Grafikformate GIF und JPEG.

CGI (Common Gateway Interface) bezeichnet eine Programmierschnittstelle, die es einem Webserver erlaubt, mit einem externen Programm zu kommunizieren. CGI-Programme sind normalerweise sehr klein und als Script oder in einer anderen Hochsprache geschrieben. Sie befinden sich auf dem Webserver und agieren als Schnittstelle zwischen einem Webbrowser und einem Content-Server (der Daten und/oder Rechenleistungen bereitstellt). Wenn ein Webbrowser die Webadresse (auch als URL oder Uniform Resource Locator bezeichnet) eines CGI-Programms aufruft, wird dieses vom Webserver ausgeführt. Meist geht es dabei um eine Sitzung mit einem Datenbank- oder Anwendungs-Server, der wiederum den Request vom Client ausführt und das Ergebnis über den Webserver an den Browser zurückgibt. Rein CGI-gestützte Anwendungen erfordern wiederholte Interaktion mit dem Server oder den Servern, um Interaktivität zwischen Anwender und Anwendung zu realisieren.

Java ist eine objektorientierte Programmiersprache, die von Sun Microsystems entwickelt und gefördert wird. Sie erlaubt es Programmierern, Web-gestützte Programme zu entwickeln, die als Java Applets bezeichnet werden. Dabei setzt sich ein Java Applet jeweils aus zwei Bestandteilen zusammen:

- eine HTML-Seite mit dem Applet
- einen Satz Java Klassen mit der Programmlogik

Die HTML-Seite des Applets definiert, welches Java Applet aufgerufen werden soll (wobei eine HTML-Seite mehrere Applets enthalten kann). Wenn ein Webserver die HTML-Seite eines Applets an einen Webbrowser übermittelt, liest dieser die Applet-Informationen und fordert die entsprechenden Applet-Klassen vom Webserver an. Nachdem diese heruntergeladen wurden, wird das Applet entsprechend der Eingabe vom Anwender ausgeführt (wobei wichtig ist, daß das Applet gelöscht wird, sobald die HTML-Seite, auf der es sich befindet, verlassen wird). Ein Applet kann beispielsweise eine Verbindung zu einem Anwendungsserver herstellen, Requests zur Verarbeitung übermitteln und anschließend die Ergebnisse entgegennehmen und anzeigen. Da ein Java Applet lokal ausgeführt wird, kann es eine weitaus umfangreichere interaktive Umgebung bieten als ein CGI-Programm.

JavaBeans bezeichnet eine komponentengestützte Architekturstruktur für Java. Damit wird eine Programmierschnittstelle definiert, die es Entwicklern ermöglicht, mehrfach einsetzbare Komponenten zu schreiben, die dann überall ausgeführt werden können, wo die entsprechende Ablaufumgebung vorhanden ist, die 'Java Virtual Machine' (JVM). JavaBeans ist das Java-Gegenstück zur ActiveX-Struktur der COM-Umgebung von Microsoft.

ActiveX ist eine Windows-basierte Lösung, die die COM™-Technologie (Component Object Models) von Windows nutzt, um eine Interaktivität und Interoperabilität mit anderen COM-Komponenten und Diensten zu ermöglichen. ActiveX-Controls bilden die nächste Generation von OLE Controls (OCX) und bieten eine Reihe von Verbesserungen, die speziell dazu ent-

wickelt wurden, um die Distribution einzelner Komponenten über das Web zu ermöglichen. Sie lassen sich ähnlich wie Java Applets in HTML-Seiten einbetten.

Ein entscheidender Unterschied zwischen **Java** Applets und **ActiveX** Controls besteht darin, daß ActiveX Controls nur auf der Windows-Plattform laufen. Einmal heruntergeladen werden sie auf Dauer installiert, können auf das lokale Dateisystem des PC zugreifen und mit jedem Rechner innerhalb des lokalen Netzwerks kommunizieren, in dem sich der Anwender befindet. Java Applets haben jedoch aus Sicherheitsgründen keinen Zugriff auf das lokale Dateisystem.

JavaScript ist eine interpretierte, objektorientierte Programmiersprache, die einen einfachen Funktionsumfang bietet. Sie wird clientenseitig (zum Beispiel im Browser) eingesetzt und erlaubt die Einbindung ausführbarer Inhalte in eine Webseite.

Dynamic HTML ist die neueste Generation von HTML, die gegenwärtig vom Netscape Communicator 4 und dem Internet Explorer 4 unterstützt wird. Sie bietet entscheidende Fähigkeiten was Interaktivität und visuelle Effekte angeht.

Unter **Compute Services** versteht man die Ausführung eines SAS Programms, wobei die Ergebnisse zum Browser des Anwenders übermittelt werden. Dabei liefert der Compute Server (also das SAS Programm) einen Ergebnissatz, der wiederum bestimmte Reaktionen eines Java Applets oder des Browsers selbst auslöst. Dabei kann es sich zum Beispiel um einen HTML-Code handeln, den der Browser des Anwenders interpretieren kann, oder um einen Datenstrom in einem Datenformat, der von einem Java Applet erkannt wird.

Data Services ist eine logische Erweiterung von SQL Diensten und beschreibt die Fähigkeit mit Hilfe von SQL auf Daten zuzugreifen, Daten abzufragen und Daten anzuzeigen. Dabei wird der Data Server allein dazu eingesetzt, um die Daten abzufragen und die Ergebnisse bereitzustellen. Die erforderliche Formatierung und/oder Weiterverarbeitung der Ergebnisse bleibt jedoch Aufgabe des anfragenden Programms (zum Beispiel ein CGI-Programm, ein Java Applet oder der Webserver selbst).

Berichtsverteilung (Report Distribution) bezeichnet die Bereitstellung von Informationen für bestimmte Personen innerhalb eines Unternehmens mit Hilfe des Webs. Dabei kann es sich um statische Informationen handeln, die periodisch erzeugt werden und lediglich passiv betrachtet werden, oder um Informationen, die auf Anforderung erzeugt werden. Der Bericht läßt sich individualisieren und damit gezielt an die Anforderungen des einzelnen Anwenders anpassen, wobei diese Anpassung sowohl vom Anwendungsentwickler als auch vom Endanwender selbst vorgenommen werden kann.

Anwendungsverteilung (Application Distribution) bezeichnet die Bereitstellung von Berichts- und Analyse-Anwendungen für bestimmte Personen innerhalb eines Unternehmens mit Hilfe von HTML-Seiten oder eines Java Applets. Es ist dabei nicht erforderlich, daß derartige Anwendungen lokal auf dem Desktop des Anwenders installiert sind. Vielmehr werden SAS Softwaredienste auf einem Server bereitgestellt und auf Anfrage an den Desktop des Anwenders übermittelt.

2. Komponenten von SAS/IntrNet

SAS/IntrNet setzt sich aus einer Vielzahl unterschiedlicher Komponenten zusammen, die einzeln oder kombiniert eingesetzt werden können, um bestimmten Anforderungen einer webgestützten Lösung zur Informationsverteilung gerecht zu werden. Dabei ist entscheidend, daß sich sowohl jede einzelne Komponente für sich einsetzen läßt, als auch eine Lösung entwickelt werden kann, die mehrere Komponenten umfaßt. Die Leistungsfähigkeit von SAS/IntrNet ist der bemerkenswerte *Umfang* jeder einzelnen Komponente.

Bei den **Web Publishing Tools** handelt es sich um eine Reihe von Hilfsmitteln zur Erzeugung statischer Webseiten auf der Basis von SAS Daten und Ergebnissen. Diese Tools können in Verbindung mit bestehenden SAS Produkten eingesetzt werden, um schnell und einfach Webinhalte zu erstellen. Die Web Publishing Tools sind kostenlos erhältlich und schließen folgende Komponenten ein:

- Die **HTML Formatting Tools** sind eine Sammlung von Makros, die Ihnen erlauben, SAS Datensätze und Ergebnisse von Verarbeitungsprozeduren als HTML-Seiten darzustellen, die von Webanwendern genutzt werden können. Dabei wird gegenwärtig zwischen folgenden Formatierungshilfen unterschieden:
  - Output Formatter
  - Data Set Formatter
  - Tabulate Formatter

  In Version 7 des SAS Systems werden sich HTML-Dateien direkt über das Output Delivery System ODS erstellen lassen.
- Die **GIF-, JPEG-** und **GIF-Anamations-Treiber** erlauben die Erstellung von Internetgrafiken mit Hilfe von SAS/GRAPH®-Prozeduren. Der GIF-Anamations-Treiber ermöglicht die Kombination von GIF-Grafiken, die mit Hilfe von SAS/GRAPH®-Prodzeduren erstellt wurden, um Webseiten mit Animationen zu ergänzen.
- Das **GraphApplet** gehört zu den 'Thin Client Graphics' und ist ein spezielles, visuelles Java Applet, das besonders schlank ist, weil es keine ständige Verbindung zum Server herstellt, sondern alle erforderlichen Informationen als Parameter beim Aufruf erhält. Ähnlich wie die HTML Formatting Tools entfaltet das GraphApplet seinen besonderen Wert im Zusammenhang mit dynamischen Techniken wie dem Application Dispatcher oder htmSQL (siehe unten), die eine dynamische Parametrisierung der Grafik ermöglichen. Ein Beispiel für ein derartiges graphisches Applet findet sich in Abbildung 4. Die Parameter und Werte, mit denen festgelegt wird, was das Applet tut oder anzeigt, sind in der HTML-Seite enthalten, in der das Applet enthalten ist.

Der **Application Dispatcher** ist eine CGI-Schnittstelle zwischen SAS Software und einem Webbrowser. Damit können dynamische Anwendungen erstellt werden, die den Zugriff auf den Leistungsumfang der SAS-Software aus dem Webbrowser heraus ermöglichen. Es läßt sich auf diese Weise grundsätzlich jedes SAS Programm starten, das innerhalb einer Stapelverarbeitung (Batch Mode) ausführbar ist. Dabei muß sich der Anwender lediglich um das SAS Programm selbst kümmern. Alle Einzelheiten in Verbindung mit CGI, wie die Übertragung der Parameter aus einer HTML-Seite und der Ergebnisse an den Browser des Anwenders werden vom Application Dispatcher übernommen. Der Application Dispatcher ermöglicht die Nutzung dieser Programme durch eine beliebige Anzahl von Anwendern, ganz gleich, ob diese SAS Software installiert haben oder nicht.

htmSQL bietet eine Schnittstelle zu SAS-Daten aus einem Webbrowser heraus und ermöglicht dynamische Datenabfragen. Dabei wird ein CGI-Programm eingesetzt, das sich auf dem Webserver befindet und sowohl den Zugriff als auch die Aktualisierung eines SAS Datensatzes (einschließlich Lesezugriff auf externe DBMS) ermöglicht. Der Anwender verwendet dafür eine Eingabedatei (.hsql-Datei) mit SQL-Abfragen, die in den HTML-Code eingebettet sind. htmSQL übermittelt anschließend die Abfrage an einen SAS Datenserver (entweder SAS/SHARE® Software oder ein Scalable Performance Data Server™). Daraufhin werden die Daten gelesen und die Ergebnisse entsprechend dem HTML-Code der .hsql-Datei formatiert. htmSQL läßt sich zur Erstellung umfassender dynamischer Anwendungen einsetzen, die es dem Anwender erlauben, Berichtsdaten gezielt an ihre individuellen Informationsanforderungen anzupassen.

Der **SAS/CONNECT Driver for Java** ist eine Sammlung von Java-Klassen, die zur Entwicklung von Java Applets und Applikationen eingesetzt werden können, die mit SAS Software auf einem Server kommunizieren. Auf diese Weise ist der entfernte Zugriff auf SAS Rechenleistungen möglich. Der SAS/CONNECT Driver for Java bietet eine ähnliche Funktionalität, wie sie einem SAS Client in Verbindung mit der SAS/CONNECT Software zur Verfügung steht, außer daß diese Funktionalität über jedes Java Programm zugänglich ist und lokal keine SAS Software installiert sein muß. Das mit Hilfe des SAS/CONNECT Driver for Java erstellte Programm kann eine SAS Sitzung starten, die Verbindung zu dieser Sitzung herstellen, Datensätze erzeugen, auf vorhandene SAS Daten zugreifen, SAS Programme zur Datenanalyse starten und die Ergebnisse übertragen.

Der **SAS/SHARE Driver for JDBC™** ist eine Sammlung von Java-Klassen, zur Entwicklung von Java Applets und Anwendungen, die mit einem SAS Datenserver (entweder SAS/SHARE®-Software oder der Scalable Performance Data Server™) kommunizieren. Diese Java Programme ermöglichen es beispielsweise einem Anwender, über eine direkte Verbindung zum SAS Server mit Hilfe von SQL-Abfragen und Befehlen Daten anzuzeigen oder zu aktualisieren.

Die **SAS SQL Library for C™** ist eine Programmierschnittstelle (Application Programming Interface oder API) zur Entwicklung von Anwendungen, die mit dem SAS/SHARE-Server kommunizieren. Der **SAS ODBC Driver** ermöglicht ODBC-fähigen Windows Anwendungen den Lese- und Schreibzugriff auf lokale und entfernte SAS Dateien. Beide sind Teil von SAS/IntrNet und erlauben den offenen Zugriff auf SAS Daten. Zahlreiche Windows Webserver bieten die Funktionalität zum Zugriff auf ODBC-fähige Datenquellen.

Die **Tunnelfunktion** ('Tunnel Feature') setzt HTTP-Tunneling ein, um es Applets zu ermöglichen, über ein CGI-Programm auf dem Webserver mit entfernten Systemen zu kommunizieren. Der Grund dafür ist, daß die Java Spezifikationen als Sicherheitsmaßnahme beinhalten, daß ein Applet keine Netzwerkverbindungen mit einem anderen Rechner herstellen kann, als dem Rechner, von dem es heruntergeladen wurde. Dieses Sicherheitsmerkmal kann bedeuten, daß die Verwendung von Java Applets zu einer nicht optimalen Serverkonfiguration zwingt, da dieser Server auf dem selben Rechner wie der Webserver installiert sein muß. Dazu kommt, daß viele Firewalls verhindern, daß Applets über die Firewall hinweg kommunizieren können, was die Serverkonfiguration zusätzlich einschränken kann. Die Tunnelfunktion spricht beide genannten Konfigurationsprobleme an. Bei Java Applets, die mit SAS/IntrNet Java Komponenten geschrieben wurden, kann die Tunnelfunktion dazu eingesetzt werden, um die Einschränkungen zu umgehen, die den Ort des SAS Servers im Bezug auf den Webserver und die Firewall betreffen.

Der **MetaSpace Explorer™** ist ein Java Applet, das dem Anwender die Möglichkeit bietet, Daten zu durchforsten, die sich in einem Data Warehouse befinden. Es erlaubt den Zugriff auf die Objekte, die in diesem Data Warehouse registriert sind. Der Anwender kann 1) ein Warehouse mit Objekten durchsuchen, die nach Thema, Typ oder Besitzer organisiert sind, 2) Objekte auf dem betreffenden SAS Server lokalisieren, 3) das Warehouse nach Objekten durchsuchen, die anderen, vom Anwender spezifizierten, Kriterien entsprechen und 4) Metadaten anzeigen, die mit dem Objekt in Verbindung stehen. Der MetaSpace Explorer erlaubt es dem Anwender, selbst zu entscheiden, wie Informationen dargestellt werden und bietet eine Vielzahl von Tools zur Suche und Anzeige der im Warehouse gespeicherten Informationen. In der Ansicht *Warehouse* wird der Inhalt des Warehouses in der jeweils höchsten Strukturebene angezeigt. Dabei werden die im Warehouse definierten Themen, Besitzer und Typen aufgelistet. Außerdem besteht Zugriff auf die Suchergebnisse. Die Ansicht *Objekt* zeigt eine Liste von Objekten, die den in der Ansicht *Warehouse* spezifizierten Kriterien entsprechen. Dabei informiert ein Symbol jeweils über die Informationsart, die in dem Objekt enthalten ist,

wie zum Beispiel HTML-Dokumente, GIF-Grafiken, Word-Dokumente, PowerPoint-Präsentationen oder Excel Tabellen.

3. Komponenten von AppDev Studio

AppDev Studio ist eine Entwicklungsumgebung zur Erstellung von Client-Anwendungen für die Entscheidungsunterstützung, die die Leistungsfähigkeit eines SAS Data Warehouse nutzen und einer Vielzahl von Anwendern zugänglich machen. AppDev Studio integriert an einem Entwicklerarbeitsplatz nicht nur alle 'klassischen' SAS-Komponenten zur Anwendungsentwicklung (SAS/EIS, SAS/AF), sondern bietet mit den Bestandteilen **webAF** und **webEIS** (s.u.) insbesondere die Möglichkeit, Java-basierte Web-Client-Anwendungen zu erstellen.

Der entscheidende Vorteil von AppDev Studio ist die Nutzung serverseitiger SAS Applikationskomponenten oder Datenbestände. Mit AppDev Studio können wiederverwendbare serverseitige Komponenten gebaut werden, die sowohl von Web-Clients als auch von Full-Clients genutzt werden können. Für die serverseitige Programmierung kommt SAS/AF (genauer: SCL='SAS Component Language') zum Einsatz. Daher ist AppDev Studio als Erweiterung und nicht als Ersatz der bestehenden SAS Anwendungsentwicklungsphilosophie zu sehen.

AppDev Studio ist ein Teil von Nashville, der bereits für V6 SAS Server verfügbar ist. Für die Kommunikation zwischen Java-Client und SAS Server wird ein Bestandteil von SAS/IntrNet genutzt (SAS/CONNECT Driver for Java). In späteren Versionen von AppDev Studio werden weitere Komunikationsstandards unterstützt werden (CORBA, DCOM).

webAF (früher Jazz/IDE) ist eine projektorientierte, visuelle Java Entwicklungsumgebung, die Entwicklern die schnelle Erstellung von Applets, Applikationen oder JavaBeans (wiederverwendbare Teilkomponenten einer Web-Anwendung) zur Informationsbereitstellung ermöglicht. webAF erzeugt '100% reinen Java Code', der bei Bedarf vom Entwickler um zusätzliche Code-Segmente ergänzt werden kann. Es lassen sich Funktionen wie Enterprise Data Access, Data Warehouse und Decision Support in Lösungen integrieren, indem ganz einfach Objekte für entfernte Server per Drag and Drop in das Projekt übernommen werden. Dieser Funktionalität liegen spezielle Klassenbibliotheken zugrunde, die auch das Alleinstellungsmerkmal von webAF gegenüber marktüblichen Java Entwicklungsumgebungen darstellen. Der Entwickler ist für die Kommunikation mit dem SAS Server aber nicht auf Standardkomponenten beschränkt: er kann außerdem eigene serverseitige SAS/AF-Objekte für den Datenzugriff erstellen, beziehungsweise integrieren.

Die Remote Object Class Factory (ROCF) als Bestandteil von webAF macht SAS Funktionalität für Java Clients zugänglich. ROCF-Objekte sind Bean-kompatible Standard-Java-Klassen. Der Anwender kann die Daten und die Rechenleistung von SAS Software nutzen, indem er sie als SAS/AF-Modelle auf dem Server installiert und auf dem Client die Proxy-Java-Klassen verwendet (die mit dem Klassen-Editor in webAF oder den separaten Tools erstellt werden). Dadurch hat der Programmierer die Möglichkeit, diese entfernten Modelle wie lokale Java-Klassen zu behandeln. Ein SAS/AF Softwaremodell ist als ein Objekt definiert, das Methoden für bestimmte Operationen unabhängig von einer Benutzerschnittstelle bietet. Die gebräuchlichste Anwendung eines Modells findet sich in dem 'Model-Viewer' Paradigma. Dabei liest das Modell Daten aus einer Datenbank, die dann vom Betrachter (Viewer) passend für den Anwender formatiert und aufbereitet werden. Mit Hilfe von ROCF kann jedes bestehende oder vom Anwender geschriebene SAS/AF-Modell auf die Java-Umgebung übertragen werden. Dasbei lassen sich die Sprachgrenzen problemlos überbrükken: die Applets oder Anwendungen auf Clientseite bestehen aus in Java geschriebenen Objekten, die wiederum entfernte SAS/AF-Programmobjekte aufrufen, die in SCL geschrieben

sind. Über SCL wird wiederum das übrige SAS-System über Aufrufe oder Scripte angesprochen. Für das Java-Objekt ist das in SCL geschriebene Objekt wie jedes andere Java-Objekt. Genauso verhalten sich auch die SAS/AF-Modelle (in SCL oder C geschriebene Objekte) unabhängig von der Sprache des Aufrufers.

webEIS (früher Jazz/SWAN) ist eine Java Anwendung für Entwickler zur Erstellung anspruchsvoller Web-OLAP-Anwendungen auf der Basis vorgegebener Geschäftsobjekte. Diese Web-OLAP Anwendungen werden als JavaBeans (wiederverwertbare Java-Softwarekomponenten) realisiert und dienen der Anzeige von Daten, die auf einem SAS Server liegen. Diese Daten können als multidimensionale Datenbanken (MDDB) oder als Tabellen gespeichert sein. Zum Lesen und Zusammenfassen der Daten auf dem Server sind entweder die SAS/EIS-Klassen oder HOLAP erforderlich. Hauptziel dieser "100% reinen" Java-Technologie ist die Anzeige und die Navigation durch Unternehmensinformationen innerhalb der vertrauten Umgebung des Browsers. Ein mit webEIS erstelltes Web-OLAP-JavaBean kann bei Bedarf in andere Java-Anwendungen integriert werden; zum Beispiel in solche, die mit webAF erstellt werden.

4. Web-Technologien und Hilsmittel zur Erstellung von Webanwendungen auf der Basis von SAS/IntrNet

Das World Wide Web ist per Definition eine offene Umgebung. Der Einsatz moderner Technologien ist integraler Bestandteil webgestützter Lösungen, wobei Webbrowser und Webserver die vorgegebenen Größen sind. Beim Einsatz von SAS/IntrNet Software sollten einige weitere Technologien berücksichtigt werden. Einige davon werden nachfolgend kurz beschrieben.

Skriptsprachen, wie JavaScript und VBScript™ sind schlanke, interpretierte Programmiersprachen mit einfachen, objektorientierten Fähigkeiten. Sie werden auf Clientseite eingesetzt (z.B. im Browser) und erlauben es, ausführbare Inhalte in Webseiten einzufügen. Das heißt, Webseiten können Programme enthalten, die mit dem Anwender in Interaktion treten, den Browser steuern oder dynamische HTML-Inhalte erzeugen. JavaScript hat eine weitere Verbreitung, wird besser unterstützt und sollte daher die bevorzugte Skriptsprache sein. Wenn die Sotware-Entwickler jedoch schon sehr gut mit Microsoft Visual Basic vertraut sind und die Anwender den Microsoft Internet Explorer als Browser verwenden, kann jedoch auch VBScript eine sinnvolle Alternative sein.

Zu den entscheidenden Fähigkeiten von JavaScript gehört die Fähigkeit, Codefragmente zu definieren, die ausgeführt werden, sobald ein bestimmtes Ereignis eintritt. JavaScipt-Code kann beispielsweise eingesetzt werden, um

- vor Absendung einer CGI-Anforderung Eingaben zu überprüfen (zum Beispiel zur Überprüfung der erforderlichen Felder, der korrekten numerischen Eingaben usw.).
- CGI-Anforderungen automatisch zu übertragen, wenn ein Element aus einem Listenfeld (einer HTML-Auswahlliste) augewählt wurde.
- Eingaben des Anwenders auf der Grundlage von vorherigen Angaben zu aktualisieren, ohne daß dafür eine abermalige Anfrage beim Server erforderlich ist
- Eigenschaften von Java Applets und Plug-ins zu lesen und zu schreiben oder deren Methoden aufzurufen

Das JavaScript in Abbildung 1 stellt eine allgemein verwendbare Funktion zur Aktualisierung einer Textanzeige und einer kumulativen Liste von Variablen in 'Drill-Down' Anordnung dar. Es wird in der Beispielanwendung 'Xplore' eingesetzt, die auf Basis des Application Dispatcher entwickelt wurde. Abbildung 2 zeigt wie sich Teile der Bildschirmanzeige je nach Ein-

gabe des Anwenders verändern. Die HTML-Seite (bzw. das Formular) wird umgehend aktualisiert, ohne daß dafür irgendeine Funktion auf dem Server ausgeführt werden muß.

```
Function drill_order(from,to,label)

{ document.forms[0].elements[to].value = ' ';
  document.forms[0].elements[0].value = ' ';

  if (document.forms[0].elements[from].checked}
   { drlorder[drlorder.num] =
        document.forms[0].elements[from].value;
     drllabel[drlorder.num] = label;
     drlorder.num++;
   }
  else
   { for(i = 0; i < drlorder.num; i++)
      { if (drlorder[i] ==
          document.forms[0].elements[from].value)
        { for(j = i; j < drlorder.num; j++)
          { drlorder[j] = drlorder[j+1];
            drllabel[j] = drllabel[j+1];
          }
        }
      }
     drlorder.num--;
   }

  drlorder[drlorder.num] = ' ';

  if (drlorder.num > 0)
   { document.forms[0].elements[to].value = drllabel[0];
     document.forms[0].elements[0].value = drlorder[0];
   }

  for(i = 1; i < drlorder.num; i++)
   { document.forms[0].elements[to].value =
         document.forms[0].elements[to].value
           + '\r\n'+ drllabel[i];
     document.forms[0].elements[0].value=
         document.forms[0].elements[0].value
           + ' ' + drlorder[i];
   }
}
```

Abbildung 1: Beispiel für ein JavaScript

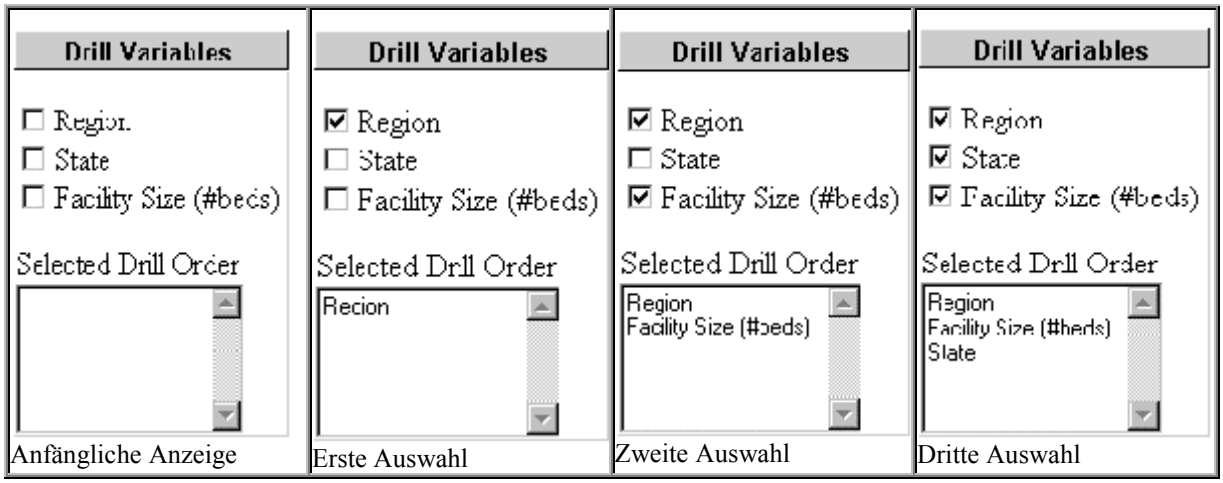

Abbildung 2: Dynamische Aktualisierung mit JavaScript

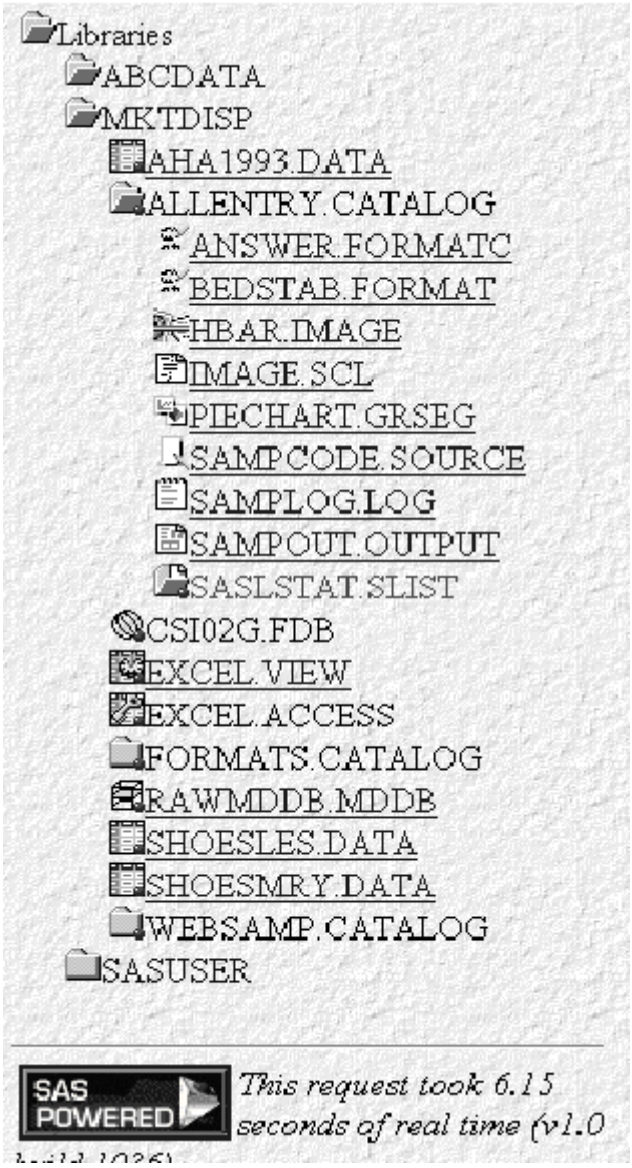

Abbildung 3: Beispiel für Dynamic HTML

Dynamic HTML hat sogar noch mehr Möglichkeiten zur Dynamisierung von Webseiten. Ein aktualisiertes HTML-Objektmodell mit maßgeblich erweiterten interaktiven Möglichkeiten wird sowohl vom Internet Explorer 4 als auch vom Netscape Communicator 4 unterstützt. Zum Zeitpunkt der Erstellung dieses Dokuments waren die Modelle von Microsoft und Netscape jedoch leider nicht voll kompatibel. Es ist zu hoffen, daß es zu einem gemeinsamen Objektmodell kommt, so daß Webentwickler Anwendungen mit diesen neuen Fähigkeiten verwirklichen können, ohne sich dabei darum kümmern zu müssen, mit welchem Browser die Webseite gelesen wird. Das Dynamic HTML Objektmodell von Microsoft scheint (nach Ansicht des Autors) robuster und flexibler zu sein und eine leichtere Implementierung von Interaktivität zu ermöglichen. Zum Beispiel lassen sich HTML-Listen, die mit der <LI...>-Anweisung erstellt wurden, mit einer einfachen JavaScript-Funktion auseinander- und zusammenklappen ('expand' und 'collapse'). Die bereits erwähnte Anwendung Xplore verfügt über einen Library-Viewer für den Internet Explorer 4 (IE4), der automatisch aktiviert wird, sobald der IE4 als Browser eingesetzt wird. Beim Laden wird

automatisch der Inhalt aller Bibliotheken und Kataloge in der HTML-Seite abgelegt. Die 'Dynamic HTML' Fähigkeit des IE 4 erlaubt dann das Auseinander- und Zusammenklappen der Bibliotheken und Kataloge in der Baumstrukturansicht, ohne daß dafür ein Verarbeitungsprozeß auf dem Server erforderlich ist (d.h. ohne Umweg über den Server). Abbildung 3 zeigt eine solche Baumstruktur. Um den für diese Funktionalität eingesetzten Dynamic HTML-Code und die JavaScript-Funktion zu sehen, kann der Leser das Xplore-Beispiel auf der Webseite http://www.sas.com/web von SAS Institute starten (unter 'Demos') und sich den Quelltext ansehen.

Ein ODBC-Zugriff auf kompatible Datenquellen ist mit vielen PC-gestützten Webservern möglich. Derartige Webserver (z.B. der IIS Server™ von Microsoft) erlauben den Zugriff auf SAS-Daten mit Hilfe der ODBC-Funktionalität, die Bestandteil von SAS/IntrNet ist.

Auch HTML-Editoren können die Entwicklung/Einführung von dynamischen Webanwendungen erleichtern. HTML-Editoren (wie z.B. Microsoft FrontPage™ oder Netscape Gold™) bieten eine visuelle Entwicklungsumgebung für HTML-Seiten. Bei diesen Seiten kann es sich um die Front-End-Menüs oder Bildschirmanzeigen der zu ertstellenden Webanwendungen handeln. Es ist möglich, mit Hilfe solcher Editoren funktionale und attraktive Webseiten zu entwickeln.

Außerdem gibt es natürlich eine Vielzahl von integrierten Java Entwicklungsumgebungen (Java Integrated Development Environments oder IDE), die sich ebenfalls für die Erstellung von Java Applets einsetzen lassen, die die Java Komponenten von SAS/IntrNet nutzen. Auch die ROCF-Komponenten lassen sich in Verbindung mit IDEs von Drittanbietern zur Entwicklung von Applets oder Applikationen einsetzen. Allerdings ist es natürlich wesentlich komfortabler, dafür webAF zu verwenden.

5. Entwicklungsperspektiven für Web-Architekturen

Im Zusammenhang mit schlanken Client-Anwendungen, bei denen ein Teil der Rechenleistung und Logik beim Client erfolgt, wird immer wieder der Begriff *Intelligenter Client* verwendet. Ursprünglich mußten derartige Anwendungen zwingend in Java geschrieben werden, da es einfach keine andere sinnvolle Alternative gab, weil CGI und HTML nicht die Interaktivität bieten konnten, die für derartige Anwendungen erforderlich ist.

Die Tatsache, daß CGI und HTML seit 1997 entscheidend weiterentwickelt wurden, während Java noch nicht so weit herangereift ist, wie viele erwartet haben, zählen heute Lösungen auf der Basis von CGI, Java und ActiveX zu den verfügbaren Technologien für die Verwirklichung wirklich intelligenter Clients.

Java ist trotzdem eine sehr bedeutende Technologie, doch seine Rolle bei dynamischen Webanwendungen entwickelt sich zusammen mit der Entwicklung des Webs selbst. Zum Beispiel werden sich JavaBeans™ (und in einem geringeren Umfang auch ActiveX als konkurrierende Technologie von Microsoft) zu bedeutenden Komponenten in Webanwendungen entwickeln, eventuell integriert in eine Dynamic-HTML-Umgebung.

Es stimmt heute nicht mehr, daß wirklich anwenderfreundliche Anwendungen und grafische Benutzeroberflächen einfach in Java geschrieben werden *müssen*, auch wenn dies 1997 noch zutraf. Java ist noch immer ein sehr wichtiges Teil im gesamten Puzzle, aber es ist nicht das einzige Teil. Vielmehr wird es viele Anwendungen geben, die allein auf Java beruhen, genauso wie es viele Anwendungen geben wird, die allein auf HTML und CGI beruhen.

Jetzt, nachdem sich CGI und HTML mit Dynamik HTML und JavaScript ergänzen lassen, um Anwendungslogik auf der Client-Seite zu implementieren, wie sie bisher nur mit Java und/oder ActiveX möglich war, kann der Begriff intelligenter Client zu Recht auch für Lö-

sungen auf der Basis von CGI/HTML verwendet werden. Welche Technologie zu empfehlen ist, hängt dabei von einer Vielzahl unterschiedlicher Faktoren ab. Die wichtigeren davon werden weiter hinten in diesem Dokument näher beschrieben werden.

Die Zukunft wird eine Komponenten-Architektur sein, in der Java, JavaScript, CGI und HTML beliebig gemischt und miteinander kombiniert werden, um gezielt die Vorteile jeder Technologie nutzen zu können. Neben dem Einsatz bei kompletten Anwendungen werden wir erleben, daß Java für kleine, schnell herunterladbare, funktionsbezogene Applets verwendet wird. Dazu gehören zum Beispiel Grafik-Applets (wie oben erwähnt) oder Applets zur Erzeugung von WHERE-Bedingungen.

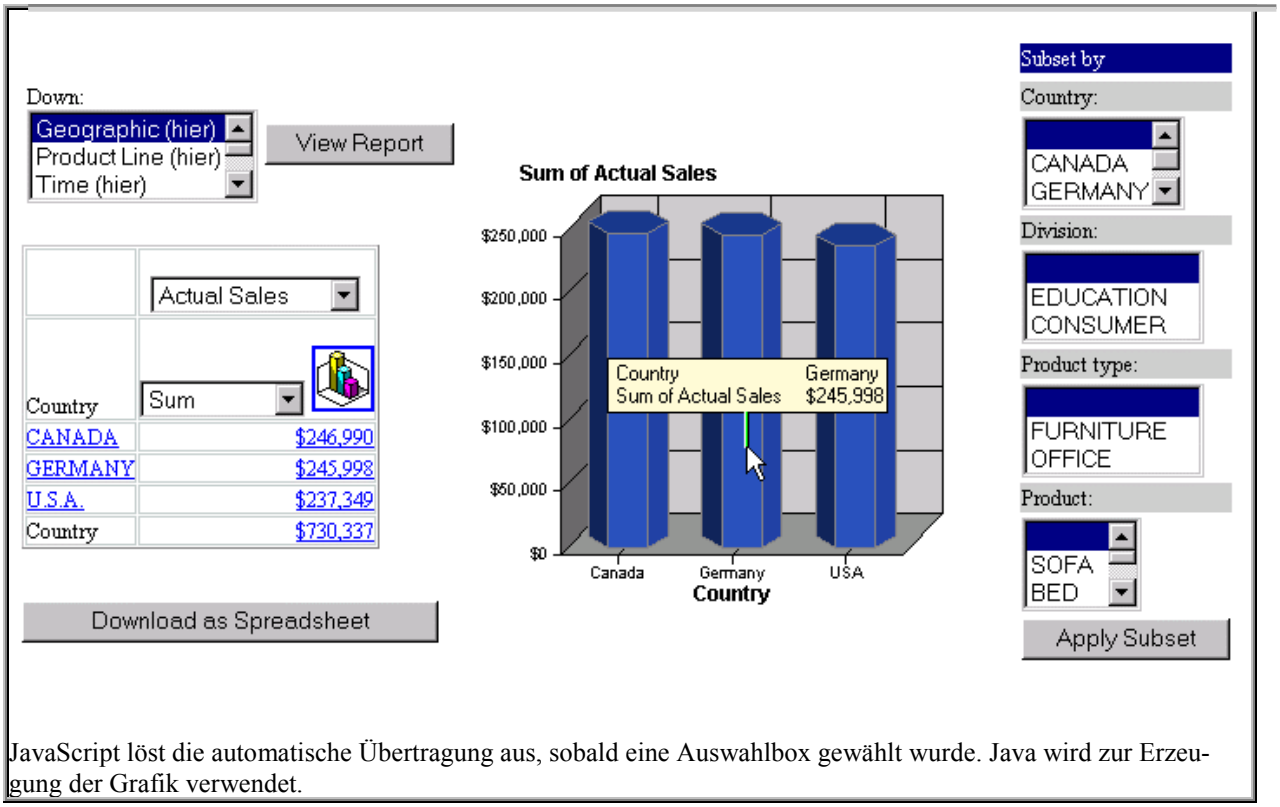

Abbildung 4: HTML-Beispielseite mit CGI/HTML, JavaScript und Java

Abbilung 4 zeigt einen simulierten Bericht, bei dem ein CGI/HTML-gestütztes Application Dispatcher Progamm für die Datenabfrage und den HTML-Code zur Erzeugung der Datenanzeige verwendet wird, einschließlich des JavaScript-Codes zur automatischen Aktualisierung der Seite, sobald eine der Zeilen oder Spalten (Produktreihe oder geographische Verteilung) gewählt wird. Das GraphApplet stellt die Grafik dar.

6. Beziehung der SAS Web-Software zu anderer SAS Software

SAS/IntrNet bietet die grundlegende Infrastruktur zur Bereitstellung der Funktionalität des SAS-Systems über das Web. Mit SAS/IntrNet kann der Anwender die Funktionalität von SAS über ein unternehmensinternes Intranet, ein Extranet oder das World Wide Web selbst einer beliebigen Anzahl von Anwendern zur Verfügung stellen. SAS/IntrNet bietet eine Architektur, die gezielt dafür entwickelt wurde, um es SAS Anwendern zu erleichtern, bestehende und neue SAS Anwendungen über das Web zu nutzen. Um Flexibilität bei der Lizensierung zu bieten besteht die Möglichkeit, für jeden Standort genau die Produkte zu lizensieren, die

benötigt werden. Es ist nicht erforderlich, die Web-Unterstützung für jeden Produktbereich (z.B. SAS/ETS oder SAS/QC) separat zu lizensieren. Stattdessen kann ein Standort SAS/IntrNet auf Serverbasis lizensieren.

SAS Web-Komponente	SAS Software auf dem Server						
	Base SAS Software	SAS/IntrNet Software	SAS/SHARE Software	SAS/CONNECT Software	SAS/EIS Software	SAS/Warehouse Administrator Software	Scalable Performance Data Server
Web Publishing Tools	X						
Application Dispatcher	X	X					
MDDB Report Viewer	X	X			X		
Warehouse Viewer	X	X	X			X	
MetaSpace Explorer	X	X		X		X	
HtmSQL	X_a	X_a	X_a				X_b
SAS/CONNECT Driver for Java	X	X		X			
SAS/SHARE Driver for JDBC	$X_{a,b}$	$X_{a,b}$	X_a	X_b			X_c
WebAF	X	X		X			
WebEIS	X	X		X	X		

Tabelle 1: Erforderliche SAS Software

X bedeutet, daß das Produkt erforderlich ist. Wenn zusätzlich ein Buchstabe angegeben wird, sind lediglich die Produkte mit demselben Buchstaben erforderlich, um die Komponente nutzen zu können.

Tabelle 1 zeigt, welche SAS Software bei Verwendung der unterschiedlichen Webkomponenten lizensiert werden muß. Dabei ist natürlich die Base SAS Software grundsätzlich erforderlich. SAS/GRAPH ist in der Tabelle nicht enthalten, ist jedoch erforderlich, wenn eine grafische Datenanzeige gewünscht wird. Die Tabelle enthält sowohl die bereits beschriebenen SAS/IntrNet-Komponenten, als auch eine Reihe unterstützter Anwendungen (z.B. der MDDB Report Viewer oder der MetaSpace Explorer). Folgende Hinweise sollten beim Lesen der Tabelle berücksichtigt werden:

- Bei einigen Komponenten gibt es mehrere Produkt-Kombinationen, die ausreichend sind, um die Komponente nutzen zu können. Zum Beispiel erfordern sowohl htmSQL als auch der SAS/SHARE Driver for JDBC einen SAS Datenserver. Bei diesem Server kann es sich entweder um die SAS/SHARE Software (einschließlich Lizenz für die Base SAS Software und SAS/IntrNet) oder den Scalable Performance Data Server handeln.

- Der SAS/SHARE Driver for JDBC erlaubt Abfragen sowohl bei einem SAS/SHARE Server als auch bei einer SAS Sitzung, die mit dem SAS/CONNECT Driver for Java gestartet wurde. In beiden Fällen ist sowohl SAS/IntrNet als auch die Base SAS Software erforderlich. Aus diesem Grund kann entweder SAS/SHARE oder SAS/CONNECT eingesetzt werden. Welche Komponente erforderlich ist, hängt davon ab, wo sich die Daten befinden oder ob ein SAS/SHARE Server aus anderen Gründen erforderlich ist.
- Die Web Publishing Tools erfordern keine Lizensierung von SAS/IntrNet. Sie können kostenlos von der Webseite von SAS Institute http://www.sas.com/web herungergeladen werden und erfordern lediglich die Base SAS Software. Zur Erzeugung von Grafiken ist natürlich auch SAS/GRAPH erforderlich.
- Der MetaSpace Explorer und der Warehouse Viewer sind Beispielanwendungen, die in Verbindung mit den Metadaten funktionieren, die vom SAS/Warehouse Administrator erzeugt und verwaltet werden. Eine Lizenz für den SAS/Warehouse Administrator ist lediglich zur Erstellung und Verwaltung dieser Metadaten erforderlich.
- ROCF und webAF werden künftig auch Corba und (vermutlich) RMI (Remote Method Invocation) unterstützen. Gegenwärtig wird zur Kommunikation mit einem SAS-Server der SAS/CONNECT Driver for Java verwendet, wofür eine SAS/IntrNet Lizenz erforderlich ist. Als dieses Dokument erstellt wurde, war über weitere Lizensierungs-Optionen (z.B. für die künftige Integration von Corba und RMI) noch nicht entschieden.

7. Entscheidungskriterien für die Technologieauswahl

Ein hauptsächliches Entwicklungsziel für SAS/IntrNet bestand darin, die Komplexität anderer Technologien, wie CGI, HTML, Java usw. abzuschirmen, die zur Einführung von Webanwendungen in Verbindung mit SAS Software erforderlich sind. Dadurch ist es beispielsweise möglich, eine Anwendung mit Hilfe des SAS/IntrNet Application Dispatcher zu erstellen, ohne daß der Anwender CGI-Kenntnisse benötigt. Allerdings gibt ein gewisses Mindestwissen über ergänzende Web-Technologien dem Entwickler eine größere Flexibilität bei der Einführung einer Weblösung. Dabei sollten jedoch Faktoren wie Anwendung und Datenverwendung für eine Entscheidung über die zu verwendende Technologie ausschlaggebend sein. Eine CGI/HTML-Lösung muß das Maß an Interaktivität bieten, das der Anwender benötigt. Andernfalls sollte dieser Weg nicht gewählt werden. Genauso wie es nicht das Ziel sein sollte, ein komplexes Java Applet einzusetzen, wenn dieselbe Lösung auch mit einer anderen (einfacheren) Technologie erreicht werden kann. Die jeweils optimale Technologie hängt von einer Reihe von Faktoren ab.

SAS/IntrNet Component	Required Skills
Application Dispatcher	• SAS programming skills - Data Step, Macro or SCL. • minimal HTML skills for interactive applications
htmSQL	• HTML • SQL
SAS/SHARE Driver for JDBC	• HTML (minimal - to build the applet HTML page) • SQL • Java
SAS/CONNECT Driver for Java	• HTML (minimal - to build the applet HTML page) • SAS programming skills - Data Step, Macro or SCL. • SQL skills may be required if making JDBC requests to the SAS/CONNECT session. • Java

Tabelle 2: Minimalkenntnisse für den Einsatz der unterschiedlichen SAS/IntrNet-Komponenten

Leider gibt es keine alles umfassende Checkliste, aus der Sie exakt entnehmen können, welche Technologien Sie für Ihre Anwendung oder ihr Berichtswesen berücksichtigen sollten. In Tabelle 2 finden Sie einige der Fähigkeiten, die für die Grundkomponenten von SAS/IntrNet erforderlich sind.

Eine Reihe unterschiedlicher Faktoren und Kriterien müssen berücksichtigt werden, wenn es um die Entscheidung für die geeigneten Komponenten geht. Dabei dürfen nicht nur die Anforderungen und Fähigkeiten berücksichtigt werden, die für die aktuelle Anwendung erforderlich sind, sondern auch andere webgestützte Anwendungen und Fähigkeiten innerhalb des Unternehmens.

Zu den Faktoren, die berücksichtigt werden müssen, gehören:

1. Gibt es bestehende SAS Programme, deren Ausgaben für einige oder alle Anwender innerhalb Ihres Unternehmens zugänglich gemacht werden müssen?

Wenn ja, dann sollten folgende Optionen in Betracht gezogen werden

- Verwendung der Web Publishing Tools zur Erzeugung von regelmäßigen statischen Berichten mit Hilfe von Batchverarbeitung
- Verwendung des Application Dispatcher, um diese Programme dynamisch nach Benutzereingabe zu steuern
- Verwendung von htmSQL für SQL-gestützte Berichte
- Bei Frontends mit komplexen und hochgradig interaktiven Menüs sollten Dynamic HTML, JavaScript oder Java Applets in Betracht gezogen werden
- Falls in den bestehenden Programmen von den objektorientierten Fähigkeiten von SAS/AF Gebrauch gemacht wird, ist die Einbeziehung des vorhandenen Programmcodes mit webAF und ROCF eine Möglichkeit
- Bei bestehenden SAS/AF Anwendungen, die auf der Grundlage einer Model-Viewer-Architektur entworfen wurden, ist die Übertragung auf das Web deutlich einfacher

Bei Ihrer Entscheidung für eine dieser Optionen sollten Sie berücksichtigen, daß es zwar eine Reihe einfacher Anwendungen für JavaScript gibt, diese Sprache jedoch nicht allzu leicht zu erlernen ist. Zum Lernen von Java muß sogar ein noch größerer Zeit- und Schulungsaufwand einkalkuliert werden.

2. Handelt es sich um eine neue Anwendung oder um eine Erweiterung einer bestehenden SAS Anwendung?

Falls wesentliche Elemente einer Anwendung neu entworfen werden müssen, ist der vorhandene Kenntnisstand der Entwickler ein entscheidendes Kriterium.

3. Werden die Berichte aufgrund statischer Daten erzeugt oder ändern sich diese Daten häufig?

Wenn es sich um statische Daten handelt und die Berichte nicht auf die Anforderungen unterschiedlicher Anwender abgestimmt werden müssen, ist vermutlich Web Publishing der richtige Weg. Andernfalls kann eine dynamische Anwendung erforderlich sein.

Wenn es sich bei dem Berichtsformat um einfache Tabellen handelt, die mit SQL erzeugt werden können, ist vermutlich entweder htmSQL oder der SAS/SHARE Driver for JDBC die beste Alternative. Allgemein ist es schneller, SQL-Requests an einen Datenserver zu schicken als ein Programm ablaufen zu lassen, das entweder den Application Dispatcher oder den SAS/CONNECT Driver for Java nutzt.

4. Gibt es feste Ausgabeformate oder muß jedem Anwender die Möglichkeit ingeräumt werden, genau die Ausgabe zu erhalten, die er benötigt?

Dabei geht es um zwei Fragen. Die erste ist, ob die Berichte mit Hilfe eines Batch-Prozessors erzeugt und einfach über das Web publiziert werden können (wofür die Publishing Tools ausreichen würden), oder ob die Berichte individualisiert werden müssen, wofür eine dynamische SAS/IntrNet Anwendung erforderlich ist.

Die zweite Frage ist das genaue Layout des Berichts. Zahlreiche dynamische Berichte lassen sich problemlos mit Hilfe des Application Dispatcher und der Web Publishing Tools (im Server-Modus) erzeugen. Wenn jedoch eine umfassendere Steuerung des *Layout* der Berichte und Ausgaben erforderlich ist, sind einige Kenntnisse über HTML erforderlich, wenn entweder der Application Dispatcher (mit individuellen DATA Steps unter Verwendung des PUT-Befehls) oder htmSQL eingesetzt werden.

5. Was für eine Art von Datenverarbeitung ist erforderlich?

Wenn es um einfache Datenabfragen in Verbindung mit tabellarischen Anzeigen geht, sollte htmSQL die erste Wahl sein. Bei Anwendungen, die eine komplexe Datenverwaltung oder Datenanalyse erfordern, sollte als Erstes einer der Compute Services (z.B. Application Dispatcher oder der SAS/CONNECT Driver for Java) in Betracht gezogen werden.

6. Welche Erfahrung ist in Ihrem Unternehmen vorhanden?

Wenn keine oder geringe Erfahrung mit CGI, HTML, Java etc. vorhanden sind, dann sollte der Application Dispatcher in Verbindung mit den Web Publishing Tools die erste Wahl sein.

Wenn die Mitarbeiter in erster Linie mit der SAS Programmierung vertraut sind, sollten Sie sich als Erstes mit dem Application Dispatcher auseinandersetzen. Wenn Erfahrungen mit SQL und HTML vorhanden sind, ist htmSQL eine attraktive Option.

Beachten Sie, daß HTML relativ leicht erlernbar ist, im Gegensatz zu JavaScript und Java. Eine minimale Investition in eine HTML-Schulung kann daher die für Sie interessanten Optionen deutlich erweitern.

JavaScript zu erlernen ist um ein Vielfaches schwieriger. Allerdings werden Programmierer, die bereits einige Erfahrung mit objektorientierter Programmierung besitzen, das Erlernen von JavaScript als etwas einfacher empfinden. Java zu erlernen erfordert eine größere Investition in Training-Ressourcen und Zeit. Allerdings kann die höhere Interaktivität und Flexibilität, die JavaScript und Java bieten, den Aufwand für Schulung oder entsprechende Personalakquise mehr als wettmachen.

7. Welche Kenntnisse lassen sich aufgrund bestehender Fähigkeiten zügig erlernen?

HTML läßt sich ohne Vorkenntnisse innerhalb weniger Tage bis zu einem zufriedenstellenden Niveau erlernen.

Programmierer, die bereits mit objektorientierte Programmierung und Syntax vertraut sind, werden Java oder JavaScript leichter erlernen können. Programmierer mit guten Kenntnissen in C oder C++ sollten in der Lage sein, Java innerhalb kurzer Zeit zu erlernen.

8. Welcher Zeitrahmen ist verfügbar?

Wenn ausreichend Zeit zur Verfügung steht, können Sie die Alternative ins Auge fassen, die erforderlichen Technologien zu erlernen, während gleichzeitig alternative Technologien (z.B. der Application Dispatcher zur Webanbindung Ihrer bestehenden SAS-Programme) als Zwischenlösung in Betracht kommen.

9. Welcher Grad an Interaktivität ist erforderlich?

Reine CGI/HTML-Lösungen werden keine unmittelbare Interaktivität erlauben, da jeder Request erst über den Server laufen muß. Wenn ein hohes Maß an Interaktivität erforderlich ist, sollten JavaScript, Dynamic HTML oder Java in Betracht gezogen werden.

10. Wie häufig soll die Anwendung eingesetzt werden?

Wenn die Anwendung nur gelegentlich eingesetzt wird, um bestimmte Berichte zu liefern, machen CGI/HTML-Lösungen Sinn, die ein schnelles Herunterladen ermöglichen und die Hauptarbeit dem Server überlassen. Wenn die Anwendung jedoch jeden Morgen als Erstes gestartet wird und den ganzen Tag über (oder über eine erhebliche Zeitspanne) genutzt wird, ist die zusätzliche Ladezeit zum Herunterladen eines größeren und anspruchsvollen Java Applets sinnvoll.

11. Wo sind die Daten gespeichert?

Die Daten müssen sich natürlich an einem Ort befinden, auf den Ihre Anwendungen zugreifen können. SAS/SHARE bietet neben der Steuerung des schreibenden Datenzugriff mehrerer Anwender den Zugriff über Plattformgrenzen hinweg. Daher können zum Beispiel der Application Dispatcher oder htmSQL ebenfalls plattformübergreifend auf Daten zugreifen, falls SAS/SHARE auf dem Rechner installiert ist, wo sich die Daten befinden, und falls TCP/IP vorhanden ist.

Wenn Sie Java Applets einsetzen, müssen die Daten entweder auf dem Webserver liegen oder es muß die Tunnel-Version entweder des SAS/CONNECT Driver for Java oder des SAS/SHARE Driver for JDBC eingesetzt werden, um die Daten auf einer anderen Plattform zu erreichen.

12. Wie groß sind die Datenbestände?

Dabei handelt es sich um eine entscheidende Frage, denn Sie sollten sicherstellen, daß unabhängig von der verwendeten Technologie die zu übertragende Datenmenge auf dem Server soweit wie möglich reduziert wird. Die Ladezeiten für große Dateien kann einen entscheidenden Einfluß auf die Systemleistung haben.

Zusätzlich zur eigentlichen Ladezeit großer Dateien kommt noch die Tatsache, daß viele Browser nicht mit großen HTML-Dateien umgehen können - besonders wenn es sich um Dateien handelt, die als Tabellen formatiert sind. Der Grund dafür ist, daß HTML-Dateien vollständig eingelesen werden müssen, bevor sie angezeigt werden können. Eine große Tabelle (beispielsweise mit Tausenden von Zeilen) kann daher leicht dazu führen, daß sich der Browser aufhängt. Wenn es erforderlich ist, einen (nicht allzu) großen Datensatz als HTML-Tabelle anzuzeigen, läßt sich das erwähnte Problem durch die Verwendung separater Tabellen für Untermengen der Daten umgehen. Der HTML Data Set Formatter (eines der Web Publishing Tools) unterstützt die gruppenweise Verarbeitung (BY-Befehl) zur Erstellung separater HTML-Tabellen.

13. Wissen Sie genau, welcher Browser zum Zugriff auf Ihre Webanwendung verwendet werden wird?

Eine einfache CGI/HTML Anwendung wird vermutlich auf jedem Browser laufen. Wenn Sie sich jedoch entscheiden, JavaScript, ActiveX, Dynamic HTML oder Java (Version 1.02 oder 1.1) einzusetzen, sollten Sie Kenntnis von den eingesetzten Browsern haben, da nicht jede Version der genannten Technologien von jedem Browser unterstützt wird. Eine Lösung kann hier das Java Plug-in der Firma Sunsoft sein, das eine Browser-unabhängige Ablaufumgebung für Java Applets darstellt.

Schlußbemerkungen

Die webunterstützenden Technologien von SAS Institute berühren alle Bereiche der SAS Software, einschließlich der End-to-end-Warehousing-Lösung. Von den Analysewerkzeugen wie der SAS Data-Mining-Lösung bis zu OLAP, Datenabfrage, Berichtswesen, und fertigen Komplettlösungen erlauben SAS/IntrNet und AppDev Studio jedem Nutzer mit einem Desktop-Computer und einem Standardbrowser den Zugriff auf die bewährten Werkzeuge der SAS Software für die Entscheidungsunterstützung. Dieses Dokument beschrieb die Technologien,

die in SAS/IntrNet und AppDev Studio enthalten sind. Außerdem wurde diskutiert, welche der Komponenten von SAS/IntrNet sich am besten eignen, um den spezifischen Anforderungen des Anwenders, beziehungsweise des Unternehmens, gerecht zu werden.

Referenzen

Henderson, Donald J. (1998), *SAS/IntrNet™ Software: A Roadmap*, SUGI 23 (SAS User Group International), Cary, NC: SAS Institute Inc.

Flanagan, David (1997), *JavaScript: The Definitive Guide*, Second Edition, Sebastopol, CA: O'Reilly & Associates, Inc.

Isaacs, Scott (1997), *Inside Dynamic HTML*, Redmond, WA: Microsoft Press.

SAS Institute Inc. (1998), *SAS/IntrNet Software: Delivering Web Solutions*, Cary, NC: SAS Institute Inc.

Umfassende Information zur SAS Web-Technologie finden Sie auf der Webseite http://www.sas.com/web.