AppDev Studio – A Roadmap

Anton Fuchs
Product Manager AppDev Studio
SAS International
Define the different types of applications used in the IT industry today and explain how AppDev Studio™ can be used to develop such applications.
Learning objectives

As a result of this presentation you should be able to:

- Differentiate the different types of applications
- Understand the benefits and drawbacks
- Know the components in AppDev Studio that map to the application types
Presentation Agenda

- Introduction
- The application landscape – The Road
- AppDev Studio components – The Map
- AppDev Studio version 3 – The Future
Introduce the topic

What is AppDev Studio?

AppDev Studio is a complete standalone development environment to create various types of applications that leverage the SAS system.
Introduce the topic

- Client/Server
- DCOM
- CORBA
- JSP
- Interactivity
- Standalone
- ActiveX
- Browser
- WML

Copyright © 2002, SAS Institute Inc. All rights reserved.
Introduce the topic

Application landscape

<table>
<thead>
<tr>
<th>Standalone</th>
<th>Client/Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCOM</td>
<td>JSP</td>
</tr>
<tr>
<td>CORBA</td>
<td>Browser</td>
</tr>
<tr>
<td></td>
<td>Interactivity</td>
</tr>
<tr>
<td></td>
<td>ActiveX</td>
</tr>
<tr>
<td></td>
<td>WML</td>
</tr>
</tbody>
</table>

Map

<table>
<thead>
<tr>
<th>SAS/AF</th>
</tr>
</thead>
<tbody>
<tr>
<td>webAF</td>
</tr>
<tr>
<td>webEIS</td>
</tr>
<tr>
<td>iPage</td>
</tr>
<tr>
<td>IOM</td>
</tr>
</tbody>
</table>

AppDev Studio
Application landscape – the road

- Standalone applications
- Client/Server applications
- Browser based applications
  - Automatic client install
  - No client install
- Wireless applications
Standalone applications

- Are installed on your local hard-drive once
- Uses local CPU resources only
- Normally makes use of local files only
- Example: MS Word
Standalone applications

- Advantages
  - No network connection needed
  - Can use all resources of the local machines

- Disadvantages
  - Difficult to support and maintain
  - All data have to be locally (redundancy)
  - Missing resources, e.g. compute power
Client/Server applications

- Are installed on more than one machine
- Local machine (client) as service requestor
- Remote machine (server) service provider
- Workload is distributed
- Resource intensive processes run on powerful machines
- Examples: DCOM, CORBA, RMI, TCP/IP
Client/Server applications

■ Advantages
  ■ Make use of powerful machines
  ■ Avoid copies of redundant data

■ Disadvantages
  ■ Network connection needed
  ■ Difficult to keep local/remote apps in sync.
Browser based applications – automatic client install

- Only run in a web browser
- Stored on a central server
- Downloaded at execution time
- Often distribute processing to a back-end server using TCP/IP, RMI, etc.
- Examples: Java applets, ActiveX controls
Browser based applications – automatic client install

- **Advantages**
  - Provide a great deal of interactivity
  - Less round trips to the remote server
  - Easy to deploy

- **Disadvantages**
  - Start up might be slow due to download
  - Needs bandwidth
  - Connection to Inter- or Intranet needed
Browser based applications – no client install

- Only run in a web browser
- Stored and run on a central server
- Normally sent mark-up back to the browser
- Examples: CGI, JSP, ASP, PHP
Browser based applications – no client install

- Advantages
  - Very easy to deploy
  - All you need is a browser

- Disadvantages
  - Lack of interactivity
  - A lot of server round-trips
  - Connection to Inter- or Intranet needed
Wireless applications

- Wireless devices (PDAs, mobiles)
- Many of the techniques mentioned before are used
- Small application foot-print on device
- Examples: WAP, IMODE, J2ME
Wireless applications

■ Advantages
  ■ Access from anywhere at anytime
  ■ No need for a wired connection

■ Disadvantages
  ■ Lack of band-width
  ■ Small devices and therefore small screen, memory, etc.
Application landscape – the road

Application avenue

Standalone

Browser based

Automatic client install

No client install

Client/Server

TCP/IP

CORBA

DCOM

ActiveX

Applets

ActiveX

Web Services

.Net

J2EE

Application avenue

Wireless

WAP

Imode

J2ME
AppDev Studio components – the map

- webAF
- webEIS
- SAS/IntrNet
- SAS/Integration Technologies
- SAS/Connect and SAS/SHARE
- SAS/AE, SAS/BASE, SAS/EIS
webAF – a visual Java development environment

- Component based (JavaBeans)
- Visual drag and drop
- Wizards to simplify the development process
- Remote access to the SAS server
- Java application, applet, servlet or JSP
- J2EE certified by SUN

standalone  client/server  automatic client install  no client install  wireless
webAF – a visual Java development environment
webAF – a visual Java development environment

```xml
<sasads:iMenu id="iMenu1" title="Year">
  <sasads:iMenuItem URL="y2000.jsp" description="2000"/>
  <sasads:iMenuItem URL="y2001.jsp" description="2001"/>
  <sasads:iMenuItem URL="y2002.jsp" description="2002"/>
</sasads:iMenu>
```

Ipage beans

Nokia WAP phone

Compaq Ipaq
webEIS – an environment to build OLAP applications

- Create interactive OLAP document
- Publish the document to the web
- Either as JavaServer Page or as applet
- Point and click environment (code-free)
- Supports standard OLAP functionality
webEIS – an environment to build OLAP applications

Sales Report by Region 1993

May 10, 2002

<table>
<thead>
<tr>
<th>Product Type</th>
<th>Furniture Actual Sales</th>
<th>Office Actual Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANADA</td>
<td>$48,564.00</td>
<td>$72,456.00</td>
</tr>
<tr>
<td>GERMANY</td>
<td>$54,246.00</td>
<td>$73,158.00</td>
</tr>
<tr>
<td>U.S.A.</td>
<td>$46,085.00</td>
<td>$74,968.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Country</th>
<th>Sum</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANADA</td>
<td>$148,895.00</td>
<td>$220,582.00</td>
</tr>
</tbody>
</table>

Pie Chart

Country
- CANADA
- GERMANY
- U.S.A.
SAS/IntrNet – CGI based application development

- A way to open the SAS System to the Internet
- Brings the power of SAS to everyone’s desktop
- Running ad-hoc reports and dynamic applications
- Uses SAS data services and compute services

no client install
wireless
SAS/IntrNet – CGI based application development

Product counts

Sales figures

<table>
<thead>
<tr>
<th>Status</th>
<th>License</th>
<th>Trial</th>
<th>Nontrial</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In Euro</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*No client install*
SAS Integration Technologies – extending the SAS power…

- … to other enterprise applications
- Enables the use of Open Standards
- Provides the SAS Integrated Object Model
- Provides a Publishing Framework
- Provides Messaging Capabilities
- Support for enterprise directories via LDAP

client/server  automatic client install  no client install  wireless
SAS Integration Technologies – extending the SAS power...

- COM
  - Open Standard

- DCOM
  - Open Standard

- CORBA
  - Open Standard

- IOM Bridge
  - SAS Standard

Client/server

Automatic client install

No client install

Wireless
SAS Integration Technologies – extending the SAS power...

SAS Workspace

DataService

Libref

FileService

Fileref

Utilities

HostSystem

FormatService

LanguageService

StoredProcessService

ResultPackageService

ResultPackage

TextStream

BinaryStream

AD0/OLE DB or JDBC

HTMLEntry

DataSetEntry

FileEntry
Sub AutoOpen()
Dim s As SAS.Workspace
Dim sl As LanguageService
Dim logline As String
Dim outlines() As String
Dim outcc() As LanguageServiceCarriageControl
Dim outlt() As LanguageServiceLineType
Dim flush As Integer
Dim line As Variant

Set s = New SAS.Workspace
Set sl = s.LanguageService

sl.Submit ("proc print
data=sasuser.class; run;")

flush = 1
Do While Not flush = 0
  sl.FlushListLines 200, outcc, outlt, outlines
  For Each line In outlines
    Selection.TypeText line + vbCrLf
    If line = Empty Then flush = 0
  Next
Loop

s.Close
Set sl = Nothing
Set s = Nothing

' Move to top of document
Selection.HomeKey Unit:=wdStory
End Sub
SAS Integration Technologies – extending the SAS power...

```
Dim ws, wsm, wsStoredProcessService, wsFref, wsBinaryStream, wsStreamHelper, VisibilityProcess = 1
Set wsm = Server.CreateObject("SASWorkspaceManager.WorkspaceManager")
wsStoredProcessService.Repository = "file:D:\euranf\seugi\2002\tkc\asp example"
wsStoredProcessService.Execute "aspsample.sas", "sex=M"
Set wsFref = ws.FileService.UseFileref("b")
set wsBinaryStream = wsFref.OpenBinaryStream(0)
set wsStreamHelper = CreateObject("SASScripto.StreamHelper")
response.binarywrite wsStreamHelper.ReadBinaryArray(wsBinaryStream, 0)
wsBinaryStream.Close
wsm.Workspaces.RemoveWorkspace ws
ws.Close

%let sex=
*ProcessBody;
filename b temp;
ods html body=b (url="body.htm");
proc print data=sasuser.class;
  where sex="&sex";
run;
ods html close;
```

...
AppDev Studio version 3.0 – the future

- Continued support for Java standards
  - J2SE 1.4, J2EE 1.3, Servlets 2.3 / JSP 1.2, JDBC 2.0
- AppDev Studio 3.0 is J2EE 1.3 certified
AppDev Studio version 3.0 – the future

Enhance general development environment

- Support team based development
- Support for Microsoft's SCC interface
- Support for CVS via SCC to CVS bridge software
- Support for Jakarta's Ant tool
- Batch build and packaging
- Build file targets are automatically exposed as project tasks on webAF menus
- Improved intelligent editing
AppDev Studio version 3.0 – the future

- Support development and deployment of web applications according to the J2EE standard
  - Creation of war files
  - Easy deployment to J2EE compliant web servers

- Support development of applications that can easily be integrated into the SAS Information Delivery Portal version 2.0
AppDev Studio version 3.0 – the future

New OLAP viewer and models for the V9 Cube Server
New relational table viewers
New chart components
And many more…

Will introduce some new JavaServer Pages custom tags
For more information

- Whitepaper “AppDev Studio – A Roadmap”
- Product information on the WWW
- AppDev Studio Developer’s site
- Brochure on AppDev Studio
- Anton.Fuchs@eur.sas.com