

Modernising Your Skills: Enterprise Application Modernisation for SOA with EGL/RBDe

Bleddyn Williams
CEO , Morpheus Limited
bleddyn.williams@morpheus.co.uk

IBM Rational Software Development Conference UK 2007



What keeps me **Rational**?



The next hour of your life!

- What we do at Morpheus
- Why did we go with Enterprise Generation Language, when we do Java?
- MotorGap
- Travel Site
- From RPG to EGL
- Some viewpoints on using EGL
- What you could gain from EGL

About Me

- Started working in web development in 1994 and have been focussed mainly on the AS/400, iSeries, System I since then.
- Started Morpheus in 1997 (10 Years).
- Involved with WebSphere since V1 and WebSphere Commerce since V2 and Portal since V4.
- What keeps me rational is the solution not doing the coding!



About Morpheus



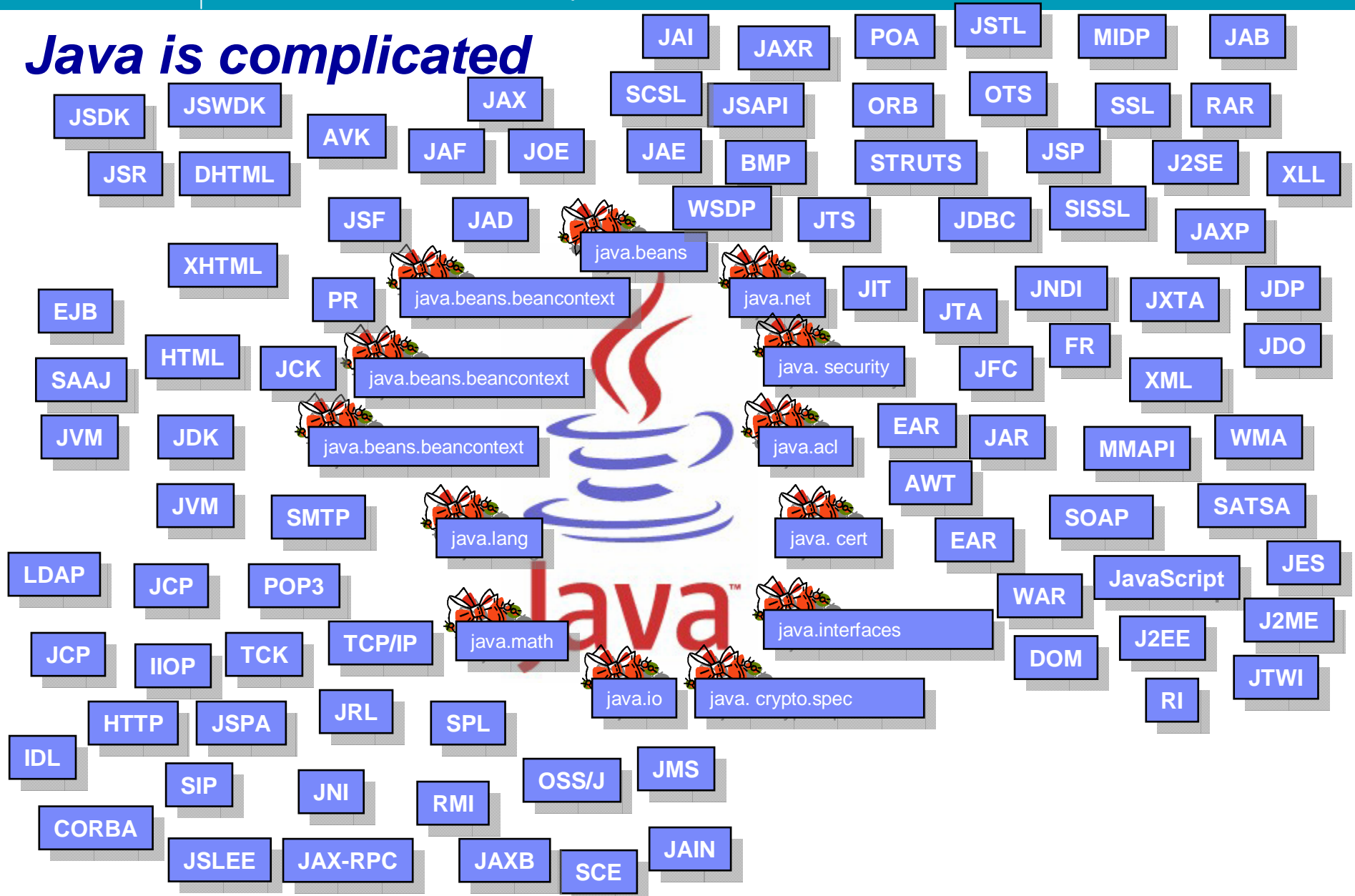
MORPHEUS
eBusiness Solutions

In the Beginning

- When we first started we used Net.Data because it was free, built into OS400 and easy to use.
- It allowed us to both build solutions for customers
- And we could reuse code they had
- After the initial hype we now build in Java



Java is complicated



So we needed an option for Customers

- That would give us the ability to create a web app in an afternoon.
- Would integrate with the existing code.
- Would make at least some sense to developers
- Then we discovered EGL the latest IBM tool for building business applications

EGL Solutions so far

- MotorGap was built from scratch there was no existing application
- Travel Insurance – migrated from Net.Data into EGL
- Helped cross train an RPG developer to build the companies first web app which is an ideal EGL environment



Please Enter Username and Password Below:

Username: -
Password:

[Click here if you have forgotten your password.](#)

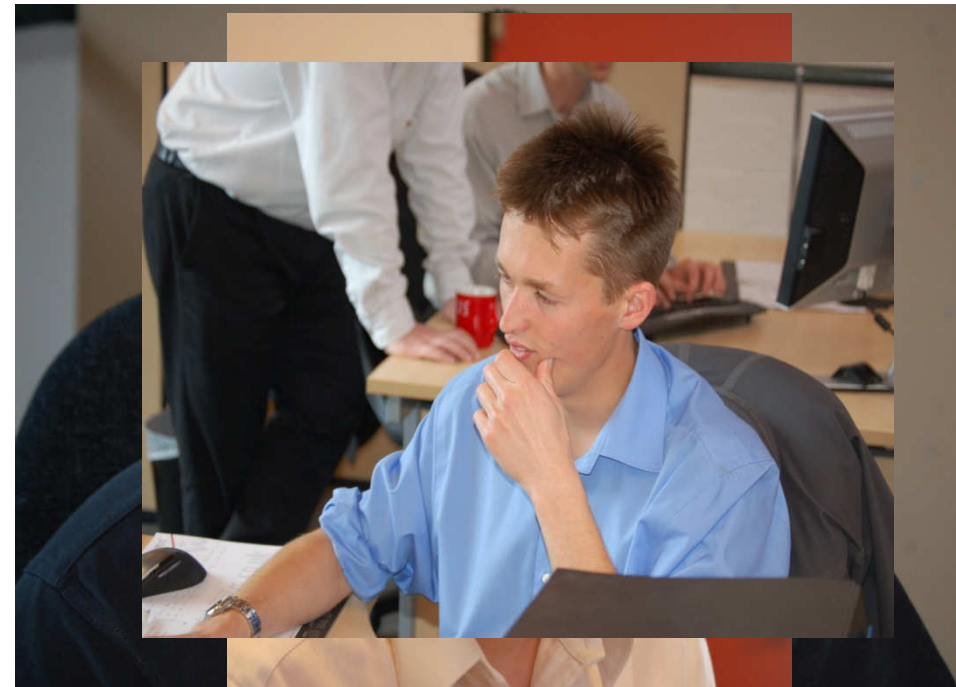
What is MotorGap and why Change?

- Gets your insurance back to when the car was new.
- Different client different options
- Competitive pressure.
- Simplicity is essential



The people who built the application

- Andy - our Java architect
- Chris an experienced Java programmer
- Lewis computer graduate
- Alex computer graduate (post implementation)
- Additional resource, as needed to get the front end tidied up sometimes me!



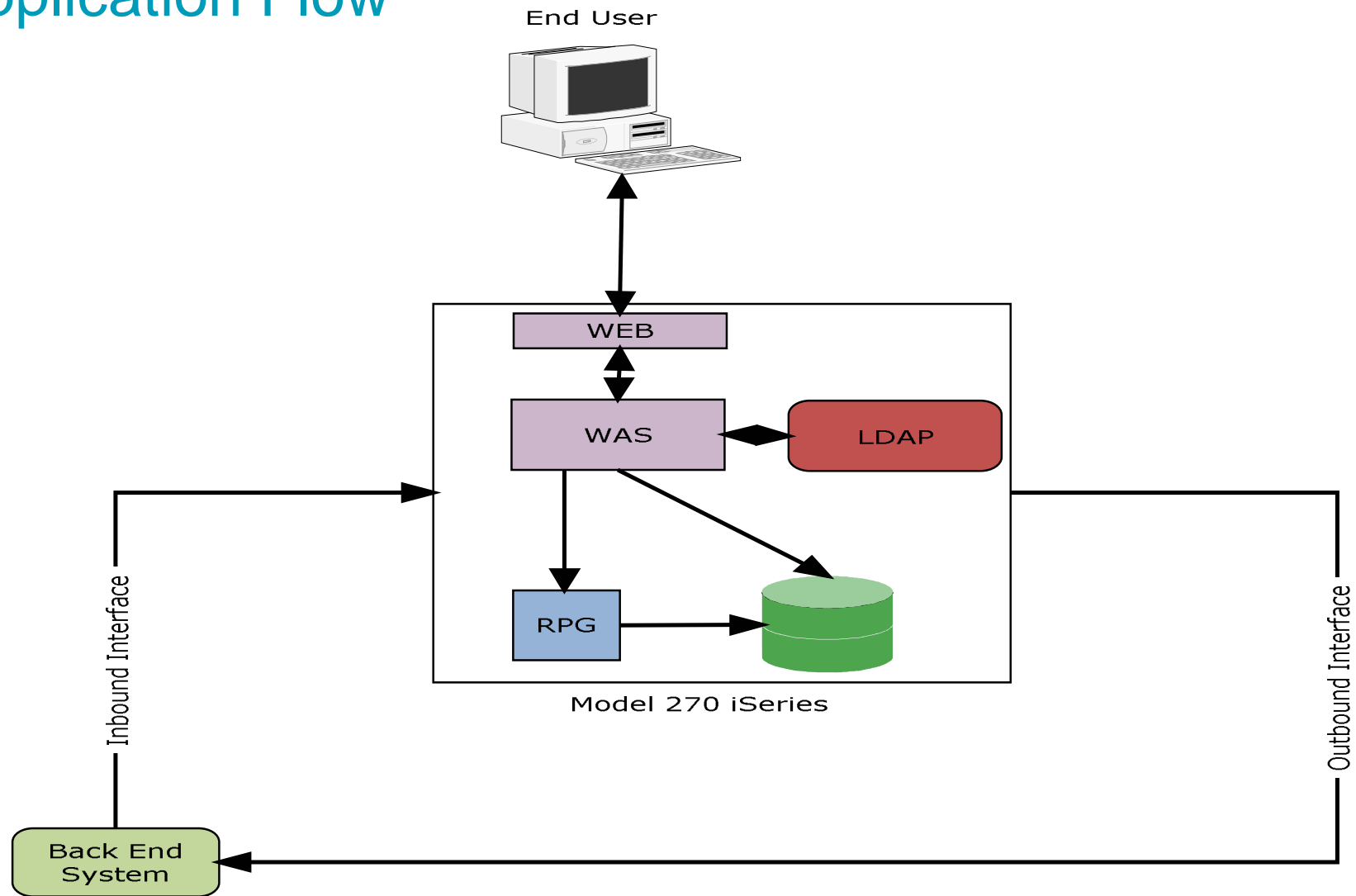
The Architecture

- The MotorGap Application was deployed into WebSphere 6.0.2.15 using a Model 270 with 8Gb memory so you can use older hardware.
- LDAP under OS400 – is being used to store user, dealer, client information.
- DB2400 being used to store the database for the application
- SQL Stored procedure provide access to existing RPG.
- Batch programs provide interfacing and run under QSH.



WebSphere. software

Application Flow



LDAP Integration

- This is a functionality that EGL did not support directly
- We provided data access to LDAP which does lookups, updates and other management functions and then as a series of Java objects and static factory methods which are wrapped with EGL interfaces.
- An EGL library then manages that integration piece populating EGL records that the rest of the application can use.
- We had to extend the LDAP schema to add our own attributes.

sn	User
mgUserRole	User
mgUserRole	DealerAdmin
isAccountEnabled	TRUE
givenName	Admin
objectclass	motorgapUser
objectclass	inetOrgPerson
objectclass	organizationalPerson
objectclass	person
objectclass	top
objectclass	motorgapEnabledUser
uid	ALP000071
mgIsAccountLocked	FALSE
cn	Admin User
mgDealeruid	ALP00007
mguserpasswordchg	11/01/2007 15:17:00
userpassword	{SHA}gX=s\p[□4□]□
mgusersecretanswer	62 75 72 6E 65 6C 6C
mgusersecretid	1
passwordflags	R

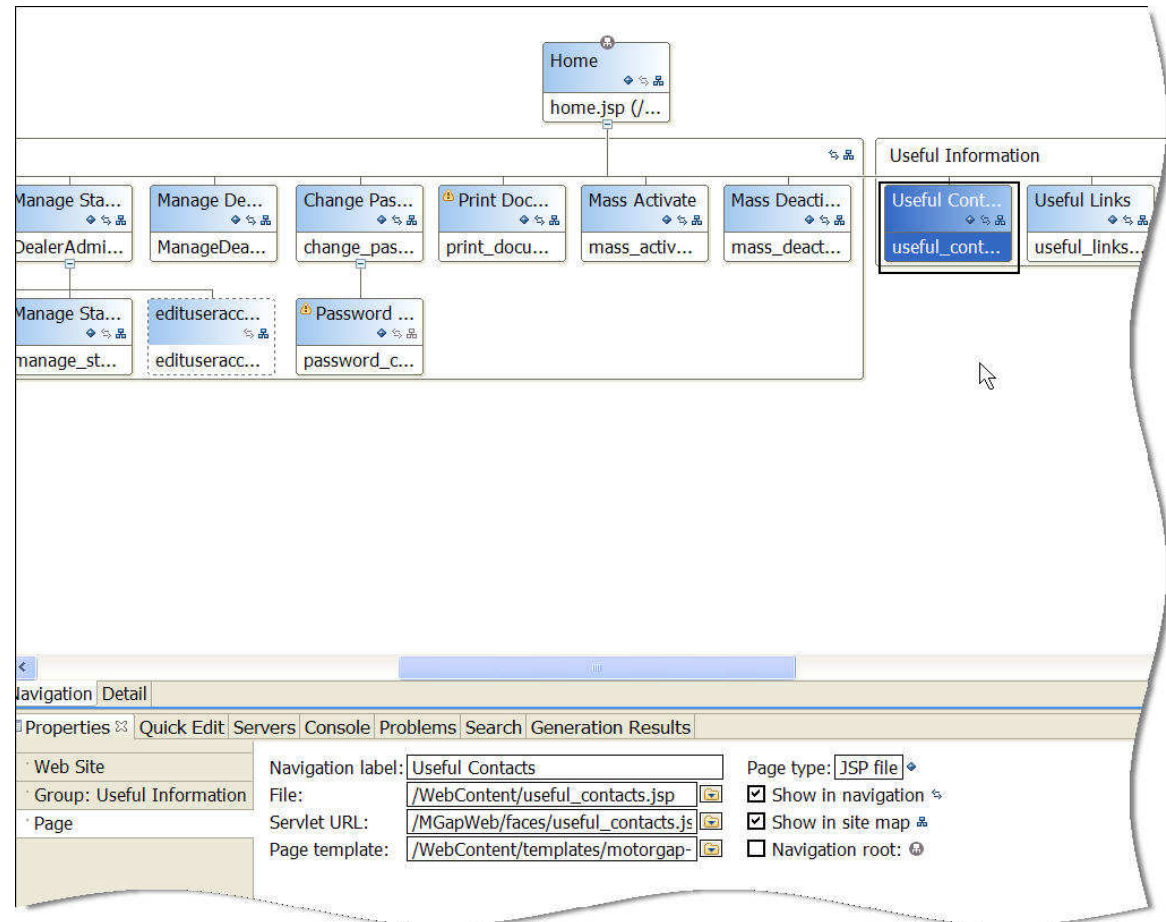
Security

- The security of the application is driven through WebSphere and the J2EE security model.
- When a request is made WebSphere intercepts the request and does the checking before it even get into EGL. All this is configured in the web.xml using security constraints.
- We have a number of servlet filters that are plugged into WAS which handle things like checking last password update and when it needs to be done (every 30 days). We then pass the responses to the EGL screens.

```
<security-constraint>
  <display-name>Public Resources</display-name>
  <web-resource-collection>
    <web-resource-name>PublicResources</web-resource-name>
    <url-pattern>/public/*</url-pattern>
    <url-pattern>/faces/public/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <description>
    </description>
    <role-name>UNAUTHENTICATED</role-name>
  </auth-constraint>
</security-constraint>
```

Dynamic Menu Items

- We use the RAD built in page navigation component to generate the layout of the site.
- We take the navigation labels in order to build the links within the site
- A good example of building on IDE functionality.



Extending Dynamic Navigation

- We have then added some Java code that looks at an additional file called the functionmap this means that
 - ▶ Websiteconfig is the structure which comes from the page navigation setup
 - ▶ Functionmap describes the roles that can see the structure

- We then render onto the screen pulling together the functionmap and websiteconfig.

```
<group label="Dealer Information">
  <role>ClientAdmin</role>
  <role>DealerAdmin</role>
  <role>HPAAdmin</role>
  <role>DealerGroupAdmin</role>
  <pages>
    <page file="view_commission_statements.jsp">
      <role>INHERIT</role>
    </page>
    <page file="view_dealer_invoices.jsp">
      <role>INHERIT</role>
    </page>
  </pages>
</group>
```

Policies
Add New Policy
Search Existing Policies
View Pending Policies
Manage Saved Policies
Dealer Information
View Commission Statements
View Dealer Invoices
Manage Accounts
Manage Staff Accounts
Change Password
Print Documents
Useful Information
Useful Contacts
Useful Links
Terms And Conditions



JSF in MotorGap

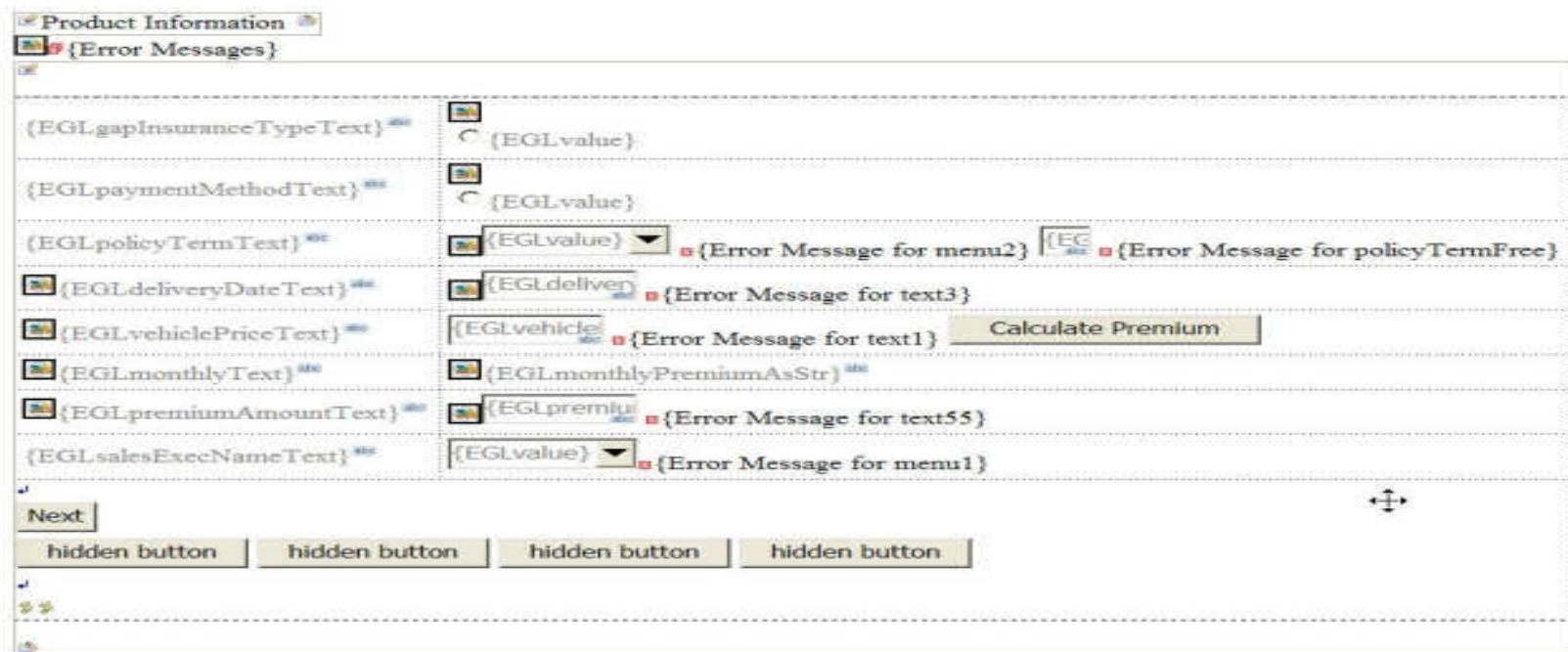
- It is very important that your team gets a good understanding of JSF and how it can integrate with EGL.
- The site although looking simple uses some complex JSF functionality take for instance the new product function.
- Like all the best websites to the user it looks simple to the developer far from it.

Product Information

Insurance Type	<input checked="" type="radio"/> Shortfall
Payment Method	<input checked="" type="radio"/> Direct Debit <input type="radio"/> Single Premium
Policy Term (e.g. 12 Months)	24 <input type="button" value="v"/>
Delivery Date (e.g. 01/04/2006)	<input type="text"/>
Vehicle Price (in £'s)	<input type="text"/> <input type="button" value="Calculate Premium"/>
Premium Amount (in £'s)	<input type="text"/>
Sales Executive's Name (if Different)	Test User01 <input type="button" value="v"/>

JSF and AJAX for the new Product Page

- We needed to be able to make things appear and disappear based on the client and choices the user makes.
- We use a mixture of programmatic changes on JSF components with EGL and others such as radio buttons combine AJAX with areas of screen for refresh.



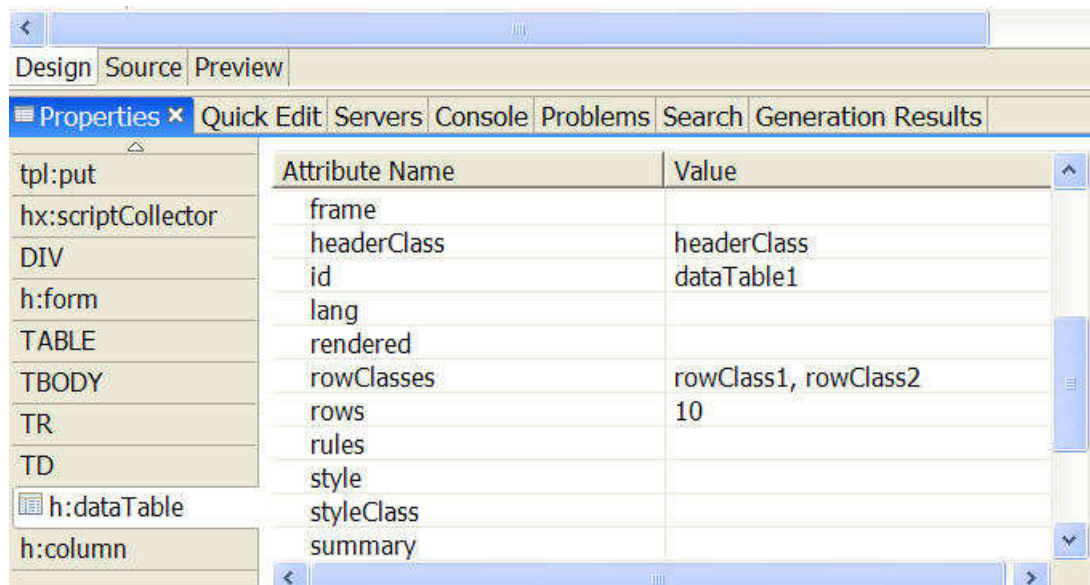
The screenshot shows a web form titled "Product Information" with a tabbed interface. The first tab is "Error Messages". The form contains several input fields and buttons:

- {EGLgapInsuranceTypeText} with a radio button and {EGLvalue}.
- {EGLpaymentMethodText} with a radio button and {EGLvalue}.
- {EGLpolicyTermText} with a dropdown menu, {EGLvalue}, and error messages: {Error Message for menu2} and {Error Message for policyTermFree}.
- {EGLdeliveryDateText} with a date picker, {EGLdeliver}, and {Error Message for text3}.
- {EGLvehiclePriceText} with a text input, {EGLvehicle}, {Error Message for text1}, and a "Calculate Premium" button.
- {EGLmonthlyText} with a text input, {EGLmonthlyPremiumAsStr}, and {Error Message for text55}.
- {EGLpremiumAmountText} with a text input, {EGLpremium}, and {Error Message for text55}.
- {EGLsalesExecNameText} with a dropdown menu, {EGLvalue}, and {Error Message for menu1}.

At the bottom, there is a "Next" button and four "hidden button" elements.

JSF and Data Tables

- It's the simple things that EGL does when working with JSF that can save a lot of time.
- A great example is colouring of rows on data tables and the simple always looks good.



Agreement	Product
WBMWD0000137	Shortfall
WBMWD0000127	Shortfall
WBMWD0000206	Shortfall
WBMWD0000199	Shortfall
WBMWD0000302	Shortfall
WBMWD0000394	Shortfall
WBMWD0000410	Shortfall
WBMWD0000450	Shortfall
WBMWD0000467	Shortfall
WBMWD0000477	Shortfall
WBMWD0000550	Shortfall
WBMWD0000550	Shortfall

Logic and Data Access

- You do not want to rip and replace existing code, one way we do this is to wrap RPG in SQL Stored Procedures.

- This does mean the RPG must be separate display (5250) from the logic, you could look at products such as Databorough to help with customers with a lot of combined presentation & logic code.

- There are 3 stored procedures which operate in this way.
 - ▶ Premium Calculation
 - ▶ Quotation Engine
 - ▶ Reference Generation



Data Access II

- When you create an EGL datapart it hard coded the library name into the statement which you have to be careful of.
- Having the Create Read Update Delete functions when dealing with single tables is very easy to maintain.
- We were a little worried about performance through the datasource but the site has been running well for eight months.

Reporting

- We use Jasper for printable reports in PDF's.
- This takes the data from EGL and combines it into the Jasper page
- We did struggle with the reporting a lot of time was spent in getting it working and populating the data into the page.
- We also had problems with getting the styles working and laying the data out into the page.

Statement of Price		
Customer Details		
INSURED PERSON	Bleddyn L Williams	
DATE OF BIRTH	01/01/1980	
INSURED VEHICLE	SAAB 9-5 AERO	
VEHICLE REGISTRATION	RA05RXB	
POLICY NUMBER	12521A602	BMW
AGREEMENT NUMBER		
Cover Details		
COVER	Shortfall	
COMMENCEMENT DATE	01/07/2007	Start Date of Cover
TERM OF POLICY	24 Months	
MONTHLY PREMIUM	£31.67	First payment to be collected approximately 28 days from the policy start date, monthly thereafter
NUMBER OF MONTHLY PAYMENTS	12	
NET PREMIUM	£323.40	
IPT	£56.64	Rate-17.5%
TOTAL PREMIUM	£380.04	

Web Services

- There were two web services we needed to use
 - ▶ Postcode Lookup
 - ▶ HPI (car details Lookup)
- The big issue we had is that they were both Microsoft generated.
- Instead we had to resort to using a standard XML HTTP interface using JAVA code and then provided an EGL interface so that it could be access.
- As a note it does seem whatever service we looked at we always got a Microsoft service!

Microsoft

WebSphere Setup

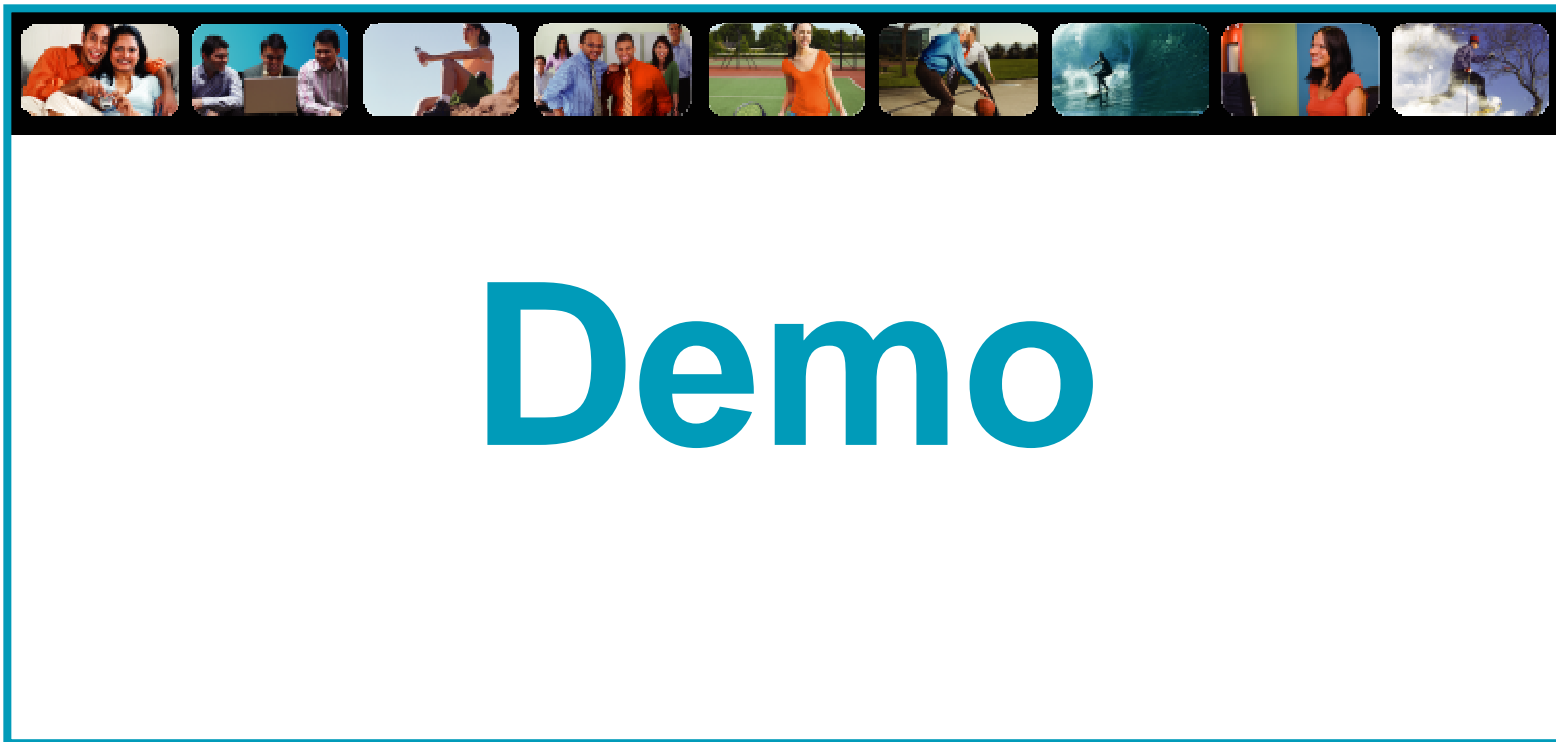
- The good thing about EGL is we could deploy it onto ANY OS (which we had to in testing as bouncing dates around is much easier on windows)
- We used V6.0.X as that is integrated into RAD 6, this made it much easier to build and test.
- Setup involved defining security with LDAP.
- Always test in your build environment we found a nice feature when we installed 6.0.2.17 that stopped EGL applications deploying.



Summing up MotorGap

- The project took around four months to build including the EGL learning curve and problem diagnosis.
- Taking many more policies per day across the different brands than the old paper system.
- Dealers get quick access to information rather than having printouts distributed to them and payments can be taken and given more quickly.
- It is being used as a key part of bids for other customers with several additional clients being added since the go live.
- Clients are looking at how else the application can be used.





The Travel Sites

- We had been running the sites for the last 8 years using Net.Data.
- The functional requirements taking the site forward meant a change.
- With our Motorgap experience it made sense to use EGL again.
- Allowed a good separation between the developer and the page builder.

Sainsbury's Bank
get a quote & apply

contact us | site map

Step 1 Year Details | Step 2 Your quote | Step 3 T&C's | Step 4 Medical questions | Step 5 Contact details | Step 6 Summary | Step 7 Purchase cover

Tell us about who is travelling and where you would like to go, and we will tell you how much it will cost.

Help and advice
Please click on the (?) icons beside each question to display additional information.

Travel Money
Order commission free foreign currency and traveller's cheques online, for next business day insured delivery to your credit card billing address.
Ready to order? [GO](#)

1 Which areas of the world will you visit? Worldwide including USA + Can

2 Enter the age band of the oldest person Less than 31 years old

3 Where did you hear about us? other

4 What is the start date of cover? 13 April 2007

5 How many adults?

6 How many children?

7 Do you require baggage/money cover? No Yes

8 Do you require cancellation cover? No Yes

9 Do you require winter sports cover? No Yes

10 Do you require additional cover for golf? No Yes

11 Do you want enhanced/reduced excess? No

http://esure-travel-insurance.co.uk - Questions - Mozilla F...

esure.com

single trip cover
Please select from the questions below.

travel details

1 Which areas of the world will you visit? Worldwide including USA + Canada

2 Enter the age band of the oldest person Less than 65 years old

3 Where did you hear about us? click or link from other site

4 What is the start date of cover? 13 April 2007

5 How long will you be away?

6 How many adults?

7 How many children?

cover details

8 Do you require baggage/money cover? No Yes

9 Do you require cancellation cover? No Yes

10 Do you require winter sports cover? No Yes

11 Do you require additional cover for golf? No Yes

12 Do you need additional business cover? No Yes

13 Do you want enhanced/reduced excess? No

Call 0845 600 3949 | contact us | terms & conditions | privacy & security | policy | FAQs | feedback

Some Travel Facts

- It took 4 weeks to re-write the application into EGL including adding new functionality.
- This included the ability to save a policy and brand the same site two or more different ways.
- Now we can manage the information a lot better before committing to the database.
- We want to integrate the application with IBM's Web Content Management system to provide ad content management.



A Customer Learns EGL

- They had a need to develop their first in house web application to allow garages to view repair information.
- They only had an RPG developer who was very good at building RPG applications but had never built a web application.
- They had existing RPG code they needed to use.
- They needed a quick turnaround and had no other resource but at the same time they wanted to learn from the development

The learning approach

- We provided an initial three day workshop to go through how EGL worked and the components involved.
- We provide any additional components which may be needed outside of EGL such as LDAP integration for security.
- The customer then worked at putting the application together and when needed we spent time helping them and answering any questions.
- After several weeks on their own they came back and we sat down for some more time with them with the finishing touches.
- We then provided onsite assistance to help deploy the application as they had never used WebSphere before.



The challenges for customers

- It was not EGL that caused the problem it is the building of a web application within a browser that causes the issues. Anyone can do a web page that says “Hello World”.
- Having to involve the PC group in setting up development environments was an interesting concept for the normally green screen based developers
- And now you need WebSphere as well to run your application

The Good

- One Integrated Development Environment
- Hides a lot of the data access complexity and the Java framework.
- Integration with Java was pretty straightforward when we needed to do things like the LDAP integration.
- The structure of the language makes it easier for RPG people to pick up.
- The support we got from the EGL team Sanjay, and Mr JSF himself Jon Sayles, Brian who enjoyed Bracknell.



The Bad

- The integration with the other RAD tooling can cause problems.
- Working in the development environment can be very slow at times when you rebuild projects. Although working with the development teams we managed to resolve a lot of the problems that did exist.
- It was confusing for Java developers at the time to get used to the syntax and the grammar being used.
- As outlined the lack of easy integration with Microsoft web services



The Not so Pretty!

- When you have to patch your development environment.
- Error messages in the runtime and development could be misleading, non existent and poor which is not a good combination
- Some of the documentation could have been improved



What can you get from EGL?

- Even as a Java company we can work with EGL using both tools to build Business Solutions
- It has given us a great tool to be able to work with customers in the System I environment.
- It gave us a new tool for new people we employee graduates who will not be experienced Java developers
- We have started using JSF now on other Java projects we have worked on rather than STRUTS
- We built on the navigation component and used it on another major project.





