UNISYS
imagine it. done.

Developing Agility

A Quarterly Newsletter for
Unisys EAE and AB Suite Customers

Contents

Not a subscriber of Developing Agility?
Don't miss the next issue – sign up in the eCommunity

Hidden Gems

By Alan Hood, Engineering Program Manager, Unisys S&T, with contributions from Gary Taylor, Ching Lam, Andy Wardle, and Nigel Tunnicliffe

Enterprise Application Environment (EAE) and Agile Business Suite provide an incredibly rich application development experience – we all know that. But as a result, there are many aspects of the products that are not commonly known. These "hidden gems" are things that you can do with EAE or Agile Business Suite that provide a big "wow" factor – with relatively little effort. And in many cases, they are accomplished by taking advantage of products or features that are already available on your ClearPath systems or other operating environments. Following are a few hidden gems we uncovered with just a little bit of prospecting.

Integrate Your "Back Office" EAE or Agile Business Suite Applications With Microsoft Products Using Web Services

Nearly every Ispec transaction interface can be generated as a Web Service, and it is very easy to integrate these Web Service interfaces with front-office productivity products, such as Microsoft® Word and Excel® using their macro scripting capability. For example, the entry of a customer number into a Word "form" could trigger a macro that runs a series of Ispec transactions to look up the customer's name and address, and automatically fill in the appropriate places in the letter. It could even retrieve account history, current balance, or other pertinent information. And, it happens transparently – Word users don't need to know that they were connected to a mainframe.

Improve Excel Integration Using the "MAPPER" Extract File Format

For applications deployed to the ClearPath environment, the format qualifier "MAPPER" or "MP" on an Extract file definition produces a flat file with a Business Information Server (formerly known as MAPPER) header and containing records with tab-delimited columns. Why is this a hidden gem? Because of how easy it is to import tab-delimited data into a spreadsheet, such as Microsoft Excel. In fact, this approach is much easier and less prone to errors than deriving the columns from the character positions in the file records. You can also use the MAPPER format to import data from tab-delimited extract files.

Tab-delimited files are more flexible than traditional position-oriented input and output. For example, when an end user changes the size of a column in the spreadsheet, there's no impact on the ability to import the data if you use the MAPPER extract file format (as long as the order of the data items remains the same, and the size of each data item does not exceed the maximum you have allowed in your report).

Provide Multiple Front Ends For a Single Application

You probably know that EAE and Agile Business Suite Client Tools give you tremendous flexibility in the way application interfaces are presented to end users. Depending on your runtime platform and the needs of your users, you can choose to use a character-based, "green screen" terminal interface, web-based Active Server Pages or ASP .NET applications, Visual Basic or Visual Basic .NET graphical user interfaces, Web Services, custom clients, and much more. Here's the hidden gem: many people don't realize that you can do ALL of these at the same time, from the same application, and with little or no change to the core application.

The secret is in the "bundle." A bundle is the collection of Ispecs that you want to include in a particular type of user interface. When you define the bundle, you copy or drag in the Ispecs that you want to have available in that interface collection and specify the name of the client generator (something like GenerateFormASPdotNET). Most developers create just one bundle and generate one client interface. But, you can have many bundles and generate many different types of clients – for different classes of users – all at the same time and without making any changes to Ispec logic. >>

Consider the case of a banking application. First, you could define one bundle that contains all Ispecs and creates a Visual Basic .NET client interface to be deployed internally for use by employees of the bank. Now, you can define another bundle that includes only the Ispecs that allow customers to log in, view account balances, and transfer funds between accounts – and specify an ASP .NET client generator for that bundle. In this way, the application automatically builds and deploys the online banking ASP .NET project, as well as the internal Visual Basic .NET clients. And, there's no need to worry about a web user accidentally wandering into the administrative functions because the bundle that you used to define their web-based client interface does not include those Ispecs.

Taking it a step further, you can offer merchants the ability to directly access your system for point-of-sale credit card validations by adding a bundle to generate Web Services interfaces for the Ispecs that handle those transactions. Again, the credit card Web Services interface is generated automatically and independently of the other client interfaces – and there's no chance of interference with the online banking or internal Visual Basic clients.

This hidden gem of deploying multiple client interfaces for the same core set of transactions can make your organization more agile and help you respond quickly to changing business needs.

Save Time and Reduce Transaction Overhead With Multiple Ispec "Instances"

Most often when developers are defining custom client interfaces, they navigate through the Ispecs using the same transaction flow they would use from a terminal or emulator – and allow the application to determine which Ispec to present. When they want to explicitly go to a different Ispec, developers use the "Page 2" function to have the current Ispec perform a Recall of the target Ispec. However, the Component Enabler interface allows you to specify the name of the Ispec directly – a hidden gem of an approach that avoids the extra transaction overhead.

The potential benefits include:
• Significantly reduced transaction load on the host – as much as 50 percent fewer calls
• Lower costs for end users when fees are paid on a "per transaction" basis
• Faster response times – typically only one Ispec is invoked, instead of two, to perform a particular function

The fine print: This technique usually only works with an application that is "stateless," or, at a minimum, in which the initial Ispec in a single transaction sequence is stateless.

What do we mean by "stateless"?
Simply put, transaction state describes the need for information about the transaction or the user to be passed from one to the next. In EAE terms, this usually means the use of GLB.WORK. If your transactions rely on data that is passed in GLB.WORK between transmits, then your application is not stateless.

Since Web Services and Service-Oriented Architecture (SOA) based applications tend to be stateless, more and more applications are being written this way. In cases of legacy applications that were originally based around terminals or internal GUI clients (i.e. where the application "knows who you are"), these can be converted to be at least partially stateless with relatively little effort.

Make "Fat" Client Distribution Easier – With ClickOnce

Today, many organizations are finding value through the use of thin client technologies, which typically leverage web browsers. One advantage is the ability to eliminate "fat" client application distribution and installation, which can require significant infrastructure investment. After all, with a browser, the end user simply clicks on the link and accesses the most current version of the application. So, browser-based approaches using ASP and ASP.NET, among other technologies, evolved as an easy way around this issue. >>

However, while browser-based technologies are continuing to expand and provide more capabilities, they may not provide the depth of features or performance of more traditional "fat" or local applications. A good example is Microsoft Outlook®, which allows web-based access but does not offer the same wealth of features or, more importantly, the speed of a local application.

Thin client/fat client – each approach has advantages and disadvantages. Therefore, it may be appropriate to offer two versions of the client: a browser-based or "thin" approach for mobile or remote users and a local or "fat" version for in-house personnel and those requiring a broader feature set.

ClickOnce – Microsoft's Visual Studio Deployment Technology
One way to streamline the process of "fat" client distribution – and one that is very easy to implement for the Client Tools Visual Basic .NET client – is the use of Microsoft's ClickOnce deployment technology. ClickOnce provides the ability to automatically distribute, download, and locally install a client application and supporting files from a web server, i.e. an http link or file server. Each time the application is initiated, it can also optionally check and download an updated version. ClickOnce addresses the challenge of "fat" client distribution and maintenance – without having to write code or employ administration staff.

Further details on the technology can be found at
http://msdn.microsoft.com/en-us/library/142dbbz4(VS.80).aspx

With ClickOnce, you get the benefits of a rich Visual Basic .NET client application with the ease of deployment of a web-based client.

Integrate EAE or Agile Business Suite With Enterprise Output Manager
While some EAE and Agile Business Suite developers may be using Unisys Enterprise Output Manager to manage, store, and distribute report output, relatively few are aware of its ability to produce nicely formatted forms that include such elements as images and barcodes. And fewer still realize that with Enterprise Output Manager, they can distribute reports via email, post report output on a protected web page, or convert it to a PDF.

Another hidden gem is the ability to integrate the Enterprise Output Manager's forms designer with EAE or Agile Business Suite Developer to create a skeleton report, and then use a special DDA (Dynamic Data Attribute) to produce truly outstanding output "forms." It's like having a Component Enabler for reports!

You may have Enterprise Output Manager – and just not know it. A limited user license is included with every ClearPath server, and many of our EAE and AB Suite customers running on Windows, Linux®, and UNIX® platforms already have Enterprise Output Manager. Check it out!

Easy Web and Mobile Output Enablement Using ClearPath ePortal Software
Perhaps you've heard about ClearPath ePortal Enterprise and ClearPath ePortal Business solutions, which are specialty engines that are used to develop and manage a variety of graphical user interfaces (GUIs) for COBOL applications in the ClearPath MCP operating environment.

Now here's the hidden gem: ClearPath ePortal can also be used to deploy and manage web-based user interfaces that have been designed with EAE or Agile Business Suite. You paint the GUI form in the EAE or Agile Business Suite painter and define the bundle for the client generation. As soon as you have generated the application, ClearPath ePortal can take over managing the IIS Web Server environment and deploy the ASP .NET or web-based mobile interfaces. ClearPath ePortal handles what can be a fairly high-maintenance process, and it lets you use the same tool to manage your EAE applications, as well as COBOL applications, on ClearPath.

This is another case where customers who already have the ClearPath ePortal product say, "I didn't know I could do that!" >>

What's Your "Hidden Gem?"

The ideas presented in this article focus on ways to get more, better, or easier use from your EAE or Agile Business Suite applications by extending the user interfaces. And, each of the techniques we've shared is being used by one or more EAE or Agile Business Suite customers – but not by the majority.

We're sure there are other hidden gems – perhaps tips for using the Development Environment more effectively or getting the most out of partial generates or testing.

Do you have a hidden gem to share? If so, please write to us at ABSuite@Unisys.com. We look forward to hearing from you.

These "hidden gems" are things that you can do with EAE or Agile Business Suite that provide a big "wow" factor – with relatively little effort. And in many cases, they are accomplished by taking advantage of products or features that are already available on your ClearPath systems or other operating environments.

Engineering Corner:
More than a Pretty Picture

We recently had the opportunity to speak with Ankur Kotwal about the development of class diagrams using Agile Business Suite. Ankur is the technical lead for Import/Export at the Australia Centre for Unisys Software (ACUS). Our goal was simple: find out more about Agile Business Suite's diagramming capabilities, including how they differ from EAE and what value they bring to the application development process. Read on to get a great overview of this powerful capability.

Developing Agility (DA): Let's start at the beginning. In Agile Business Suite, UML class diagrams replace Graphs in EAE. What is a UML class diagram? How is it different from and better than an EAE Graph?
Ankur Kotwal (AK): Unified Modeling Language – or UML – is an industry-standard modeling language used to specify, visualize, construct, and document the structure and design of an object-oriented (OO) software system. Class diagrams, which make up one of the 13 standard types of UML diagrams, show the collaborations and relationships between a collection of classes and interfaces.

An EAE Graph is a custom-built tool for visualizing an EAE system that uses non-standard notation. You can create objects in an EAE graph, assign relationships, and see when the relationships have been coded (the letter turns from lower case to upper case when the logic is added).

The Agile Business Suite UML class diagrams provide much more functionality than the EAE Graphs because they are more than just a view into the application. UML class diagrams can be used to model a system from the ground up. And, changes made to the classes in the diagram are applied to the system in real time – a potent "active diagramming" capability that is unique to the Agile Business Suite toolset.

The developer can include as much or as little of a system as desired in a given diagram. With Agile Business Suite and UML diagrams, you prototype and work at the model level – and see your change flowed immediately to the working application in your development environment.

DA: Can a class diagram help with making enhancements to an existing application – one that has been imported into Agile Business Suite from EAE? If so, in what ways?
AK: First and foremost, class diagrams allow developers to visualize the application's components and their interactions. In doing so, a user is clearly able to see the dependencies between and among classes. So, Agile Business Suite's diagrams provide a good, high-level visual overview of the migrated EAE application. Some customers have told us that they want to take advantage of OO concepts, but those efforts are made more difficult because there's no longer a developer on staff that really understands the EAE application. UML diagrams help overcome that issue by visually documenting the migrated application.

In addition, the diagrams can help developers make improvements to the application. For example, class diagrams can aid in identifying duplicated attributes in migrated EAE applications. Once found, these redundancies can be removed by creating a new class for common functionality that is reused in multiple locations. This reduces ongoing maintenance and improves the overall system design.

DA: If someone is building a new application in Agile Business Suite, should they start by drawing a class diagram? If not, how should they start?
AK: Classes in Agile Business Suite are the fundamental building blocks of all applications. Therefore, every Agile Business Suite application begins as a set of classes. Class diagrams aid developers in rapidly modeling complex classes and relationships in a diagrammatic fashion – enabling the design of a high-level model of their application in a very short amount of time. Since the AB Suite class diagram is an actual editor, all the elements created/modified in the class diagram are actually modeled in the system. >>

Keep in mind that while you are creating the model using the Agile Business Suite diagram, you are also building your application. Many other modeling tools simply create the model – and then developers have to go use other tools to build the application. But with Agile Business Suite, the creation of the model and the building of the application happen at the same time. Because of the way AB Suite builds the application directly from the model, we can say that the model is the application.

DA: Most customers are accustomed to writing their application by defining screens and using LDL constructs to define the logic. How would class diagrams change this process?

AK: Before designing screens and logic, a developer must have classes and attributes. Data entities in Agile Business Suite are modeled as persistent attributes, which are members of classes. All the classes and attributes (along with their properties, such as persistence) can be modeled through a class diagram. The class diagram does not necessarily change the sequence of application development as much as it allows the developer to complete the modeling phase faster.

DA: What are the real benefits of using a class diagram?

AK: The strength of class diagrams is in providing a rapid, visual development tool. They also make it very easy to see or visualize the design of a given application. They can be used to diagrammatically view as much or as little of the system as desired. Thus, larger systems will most likely consist of multiple class diagrams.

DA: Are class diagrams used mainly for application design? For documentation?

AK: Agile Business Suite class diagrams fulfill multiple roles. They not only aid in designing the application – they also form the basis of documentation for an application. And as I mentioned earlier, because Agile Business Suite class diagrams actually model the application itself, they provide true traceability between the design and implementation of the application. Unlike traditional modeling and development tools, the design and implementation can never be out of sync with Agile Business Suite. So, no more mismatch between code and documentation – with Agile Business Suite it's self documenting.

DA: Do class diagrams help with debugging an Agile Business Suite application?

AK: No, class diagrams are not used in the debugging stage of the application development lifecycle. That said, if a developer is trying to track down the cause of a problem, class diagrams can make it much easier to identify the elements affected by the issue and possibly provide an indication as to where the root cause lies.

DA: Do class diagrams offer value when creating a distributed or Service-Oriented Architecture (SOA) based solution?

AK: Class diagrams represent a single application – not an ecosystem. However, they can be utilized to document the external interfaces of an application, which is necessary in a SOA environment.

DA: Can class diagrams be used to plan a network environment?

AK: No, they're not suitable for modeling network interaction.

DA: What about this thing called "XMI"? How would I use other tools, like Rational RSM, with Agile Business Suite?

AK: XML Metadata Interchange (XMI) is an industry-standard notation for the exchange of UML information. Rational RSM is a popular UML modeling application that utilizes XMI as a way of interchanging UML models with Agile Business Suite, as well as other tools. As such, an application can be modeled in Agile Business Suite and transferred to RSM via XMI (and vice versa). To learn more about importing models into Agile Business Suite via XMI, read my article in an archived edition of Developing *Agility*. >>

DA: What's the hardest thing to get used to with UML diagrams when a developer is coming from EAE to Agile Business Suite?

AK: Well, Agile Business Suite's class diagram tool is very similar to other diagramming tools. So, I would encourage EAE developers to simply "give it a go." Probably the biggest challenge is a lack of familiarity with UML notations. But, it's amazing how quickly people can come up to speed once they get comfortable.

And, it's worth the time to learn more about OO and UML concepts. Developers get the most benefits from class diagrams when OO principles are applied to the application development process. Plus, you should keep in mind the benefits of the diagrams – the ability to keep the design and application completely in sync and the speed with which you develop and implement changes by working at the model level.

Providing a quick response to change has always been a hallmark of Unisys development tools – and Agile Business Suite continues that tradition. After the model is built and the application deployed, the model still provides value, because it continues to be used to implement changes and enhancements.

Many thanks to Ankur for providing an excellent overview of Agile Business Suite UML diagrams. If you have questions, please send an email to ABSuite@unisys.com*.*

Some customers have told us that they want to take advantage of OO concepts, but those efforts are made more difficult because there's no longer a developer on staff that really understands the EAE application. UML diagrams help overcome that issue by visually documenting the migrated application.

Telco Migration Case Study

We recently posted a comprehensive case study that reviews how Unisys Global Communications Industry Group migrated its Messaging Solution to Agile Business Suite. The project was a collaborative effort between Unisys teams from across the globe and the case study includes some nice detail about the migration process itself.

The Unisys Messaging Solution is used by more than 90 Telcos worldwide, including many Tier 1 operators, to deliver voicemail services to residential and business customers. In aggregate, the solution supports an estimated 200 million-plus mailboxes and more than 350 million messaging transactions daily. What's more, every installation starts with the core product and is customized to meet the needs of each customer. This means a migration is performed for every deployment.

Like many development organizations, the Global Communications group wanted to preserve the value and stability of its high-performance, feature-rich voice messaging solution while modernizing the underlying development and deployment technology. It was able to do so by moving the Messaging Solution to Agile Business Suite – and taking advantage of safe passage migration.

We hope you'll take the time to read the full case study in the eCommunity to learn more about the process and how Unisys Global Communications Industry Group was able to move development to the modern Agile Business Suite platform – while ensuring a trouble-free transition for its customers.

Key benefits include:
• Enhanced business value via the ability to gain continued return on investment from the Messaging Solution
• Continued solution stability via safe passage migration to Agile Business Suite without any noticeable interruptions to customers
• Improved productivity, including an 80% reduction in code migration elapsed time compared to a manual approach
• Enhanced long-term development agility and faster time-to-market using a modern, model-driven toolset

For tips on planning your migration to Agile Business Suite, check out the 3Q2008 issue of Developing *Agility*.

Join Us in Minneapolis this November

The 2009 UNITE Conference is scheduled for November 8-11, 2009 at the Hyatt Regency Minneapolis. And, now's the time to take advantage of one of UNITE's special registration incentive programs. There's an incentive for everyone – from loyal attendees to newbies. Check out the UNITE web site (unite.org) for all the details. Don't delay, some offers require registration by July 31, 2009.

Why attend UNITE? To learn new things and network with your peers, of course. As in previous years (and pending UNITE approval), the EAE track organizers will be providing plenty of hands-on opportunities to test drive Agile Business Suite 2.0 in a lab setting. There's just no better way to get so much information with just a few days investment of your time.

And when you come to UNITE, you are joining an organization that's 100 percent dedicated to providing a forum for Unisys users to exchange ideas and information – extending well beyond the annual conference. Let your voice be heard by submitting requirements, volunteering for an advisory council or task force, or participating in the election process. You may even consider becoming a UNITE official!

Every year, the EAE and Agile Business Suite track at UNITE has great attendance. And in light of this tight economy, there's no better value. So, make your plans today!

Calendar

There are many invaluable learning opportunities available to you. Please be sure to check the Webcasts & Events section of the eCommunity for the latest information.

What	Where	When
Technology Updates – Agile Business Suite and EAE user meetings	Amsterdam, NL Milton Keynes, UK	June 16, 2009 June 18, 2009
Agile Business Suite and EAE two-day user meeting	St. Paul de Vence, France	November 2-4, 2009
UNITE Annual Technology Conference	Hyatt Regency Minneapolis, Minneapolis, MN	November 8-11, 2009