



## Contents

### Customer Q&A: de Clarens Migrates to AB Suite

A century-old insurance brokerage is migrating to AB Suite in a Windows production environment. M. René Conte, IT director, discusses how de Clarens has planned the move.

Page 2

### Engineering Corner: Polymorphism

Polymorphism helps developers do what they like to do best – write clean, elegant code that is easy to maintain.

Page 4

### Learning a Business Perspective

As part of the curriculum at two universities, AB Suite is helping students in the former Soviet Union apply western development practices to solve global business challenges.

Page 8

### AB Suite Jump Start Workshops Continue in 2008

50+ students from 18 companies attended the sessions in 2007. Find out what you can gain from attending.

Page 9

### Annual User Conferences Attract 100s

UNITE, Future Matters, and the AB Suite Seminars provided excellent educational and networking opportunities in the second half of 2007.

Page 10

### Developing Agility en Francais? Oui!

Starting with this issue, a [French](#) translation of Developing Agility is available, joining the [Spanish](#) and [Portuguese](#) versions.

### Windows Vista Support

Starting in November 2007 with IC 3220, EAE Developer 3.3 is supported with Windows Vista. AB Suite Developer 1.2 has been supported with Windows Vista since September 2007, starting with IC1350. See the [Unisys Product Support web site](#) for more information.

### Events

Check our calendar for the latest information about 2008 plans.

Page 11

## Customer Q&A: De Clarens Migration to AB Suite

A regular feature of Developing Agility, Customer Q&A provides real-world perspectives and practical advice regarding the migration and use of Agile Business Suite (AB Suite).

This issue brings you an interview with M. René Conte, IT director for de Clarens, a century-old insurance brokerage firm headquartered in Paris, France. Like many enterprises, de Clarens depends on its IT infrastructure and development capabilities to support the growth of its diverse product and service offerings. For more than 30 years, de Clarens has partnered with Unisys for technology solutions and consulting. Our discussion with Conte describes how a recent migration to AB Suite has enabled ongoing business growth.

**Developing Agility:** Please describe de Clarens' primary business challenge and how IT supports the business.

**René Conte:** De Clarens is very focused on growing the business, which can include mergers with or acquisitions of other brokers. Our management understands that IT investment supports this expansion, especially new capabilities for integration. IT needed to find better ways to quickly join the application systems of new partner companies with our current applications.

**Developing Agility:** Please tell us about the IT environment at de Clarens.

**René Conte:** Our core business application, SARA, is a custom solution we developed to track customers, policies, guarantees, and so on. It was originally written in COBOL, and then re-written in EAE, and now migrated to AB Suite. SARA has 500 reports and Specs and is hosted on a Microsoft Windows platform. We have several other applications from the market, such as payroll, that are not written with a 4GL but do interface with SARA.

De Clarens has two full-time developers and relies on external resources to supplement the headcount, as required. We support 100+ end users.

Before migrating to AB Suite, we modernized our end-user interfaces using a custom ASP generator that was developed by Unisys France.

**“It was very important to preserve our EAE application. ... we also wanted to have a solution with excellent integration capabilities.”**

**Developing Agility:** Describe your decision-making process in choosing to move to the new development environment in AB Suite.

**René Conte:** It was very important to preserve our EAE application. And as I mentioned earlier, we also wanted to have a solution with excellent integration capabilities. EAE does provide some features, but AB Suite makes integration very easy through Visual Studio. For example, we can call C++ and C# code directly from SARA using AB Suite.

**Developing Agility:** What business benefits do you expect to gain from AB Suite?

**René Conte:** With AB Suite, de Clarens can move to a modern development environment and retain the investment in our EAE solution. In addition, we can integrate with services in other applications written using other languages very easily and rapidly. Our managers continue to ask for additional GUIs and more extensive application access from the Web. Now, with the new screen painter in AB Suite and the custom ASP generator, we can deliver these capabilities quickly and with less work.

**Developing Agility:** What AB Suite features were important influencers in your decision making?

**René Conte:** First, I was concerned about losing data and development time – so safe passage from EAE to AB Suite was extremely important. Secondly, OO features were a factor because they can help us respond quickly to requests from de Clarens management. And, I must again stress the importance of integration with other applications.

"The migration was very easy, including moving the model file to AB Suite."

**Developing Agility:** Would you tell our readers how de Clarens planned and conducted its migration?

**René Conte:** As you know, de Clarens has worked with Unisys since 1975. Five years ago, Unisys and de Clarens created a technical roadmap that we have been following ever since. We are working closely with Unisys consultants on the AB Suite migration. First, Unisys consultants converted our EAE application model file to AB Suite at the local Unisys office. Then, they installed AB Suite at our site, loaded the model, and left the migrated solution with us for one month of testing. We plan to go live in early 2008.

**Developing Agility:** How easy was the migration? What did you learn in the process?

**René Conte:** The migration was very easy, including moving the model file to AB Suite. We didn't have to do anything special and the results were very clean. So, this step was quite simple. We learned that we could migrate rapidly and without much cost.

**Developing Agility:** What have been your experiences using AB Suite in a production environment?

**René Conte:** We plan to go into production in early 2008, but during all our testing, we have seen no operational differences at all. Our end users have not seen any changes and are experiencing almost the same efficiency and performance. The ASPs that we have running with EAE are running in AB Suite without change.

**Developing Agility:** Tell us what your developers have to say about the new AB Suite toolset.

**René Conte:** Our developers attended training from Unisys about the migration and the differences between EAE and AB Suite. As you know, there is a big change in concepts from EAE to AB Suite. With EAE, we used Reports and Specs. With AB Suite, we have methods and classes. This is a real difference in mindset. For now, our developers are

using AB Suite as they used EAE. By that I mean that they are not taking advantage of all the AB Suite OO capabilities – yet. But that will change. We contracted with Unisys to work side-by-side with our developers on an informal basis for several weeks to help them learn how to use AB Suite properly.

**Developing Agility:** What are your plans for AB Suite going forward?

**René Conte:** We want to learn to work in “the AB Suite way.” De Clarens has more plans to grow the company, so we will have new solutions to integrate. All new development will be done using AB Suite.

**Developing Agility:** What advice do you have for other organizations that are migrating to AB Suite?

**René Conte:** I really recommend AB Suite to all organizations that use EAE, particularly those that have to integrate with other solutions. AB Suite makes it easy to build composite applications. I also think that you should ask for the support of Unisys consultants because their help makes the migration much easier.

**It was very nice to speak with you, René! We wish you and your team all the best as you move forward with AB Suite. For more information about de Clarens, please visit [www.declarens.com](http://www.declarens.com)**

## Engineering Corner: Polymorphism

By Howard Bell, Senior Software Engineer, ACUS, and Alan Hood, Engineering Program Manager, Unisys S&T

We've been focusing Engineering Corner on discussions related to OO concepts, including stereotypes ([2Q2007](#)) and encapsulation ([3Q2007](#)). What's next? Why polymorphism, of course! Yes, another long and unfamiliar term, but one that developers must grasp because of the significant benefits it offers. Polymorphism (along with inheritance, encapsulation, and other OO constructs) helps developers do what they like to do best – write very clean, elegant code that is incredibly easy to maintain.

### What is Polymorphism?

Polymorphism is a term commonly used in biology. If you Google it, you'll probably find a definition such as this one from the American Heritage Science Dictionary:

**polymorphism** (pŏl'ē-mŏr'fīz'əm) The existence of two or more different forms in an adult organism of the same species. In bees, the presence of queen, worker, and drone is an example of polymorphism.

In OO, the concept is similar and, as with bees, best understood through an example. Banks have various types of accounts, such as no-fee checking, standard checking, 55-and-over checking, basic savings, and money markets. And, the list continues to expand as financial institutions look for ways to differentiate themselves in the marketplace.

Fundamentally, all of these variations are different types of the Account class and have certain shared characteristics as a result, such as account number and balance. These attributes are defined once at the Account class level and inherited by each subclass. Of course, each subclass has its own unique attributes and behavior, but the idea is to strive for commonality whenever possible to simplify development, improve reusability, and reduce maintenance.

Now, these accounts accrue interest in a variety of ways. No-fee checking earns zero interest and each of the others earns at a different rate. A traditional programming approach to this is to define one subprogram to calculate the interest for all types of accounts with parameters, such as the account type, as well as various dates and financial values.

Using OO concepts, another approach would be to define the interest calculation as a method within each account type. For example, each account type could have its own CalculateInterest method. However, all accounts *do* have interest calculation in common – even if it's “no calculation.” Therefore, thanks to polymorphism, the better approach is to define the CalculateInterest method at the Account level and let the actual determination of how to perform the calculation happen at run time based upon the type of account.

Consider the following example. To begin, the “standard” CalculateInterest method is defined in the base Account class. Then, for those subclasses that are different, a local variant of the CalculateInterest method is defined, which *overrides* the method from the superclass (Account).

Polymorphism makes this possible and the syntax is as follows:

```
InterestAmount := ThisAccount.CalculateInterest()
```

With polymorphism, the statement above is written just once and applied to all types of accounts. At runtime, the system is able to determine the type of class assigned to the local variable ThisAccount. Therefore, when this statement is executed on a Savings account, the Savings version of the CalculateInterest method is invoked. When the same statement is executed on a Checking account, however, the Checking version of CalculateInterest is invoked.

Polymorphism eliminates the need for complex logic, such as Case statements or nested If statements, to get to the right result. Rather, the appropriate method is *automatically* called for the type of account being processed.

In this example, polymorphism gives the developers of the banking application important options. First, they can write one method for calculating interest that is common to most account types, which will automatically be inherited and used by any subclasses. For accounts that have different or more complex interest calculations, however, the developers can write a method with the same name specifically for that subclass and have it override the method from the Account superclass. So if the bank decides to offer a new account with special interest rates for young adults from the ages of 21-29, there's no need to change the ForEach loop, or the statement that invokes CalculateInterest. Simply create a new Account subclass with its own interest calculation and there you have it!

## **Behind the Scenes: Database Design to Support Polymorphism**

One interesting aspect of polymorphism in AB Suite is the physical implementation at the database level.

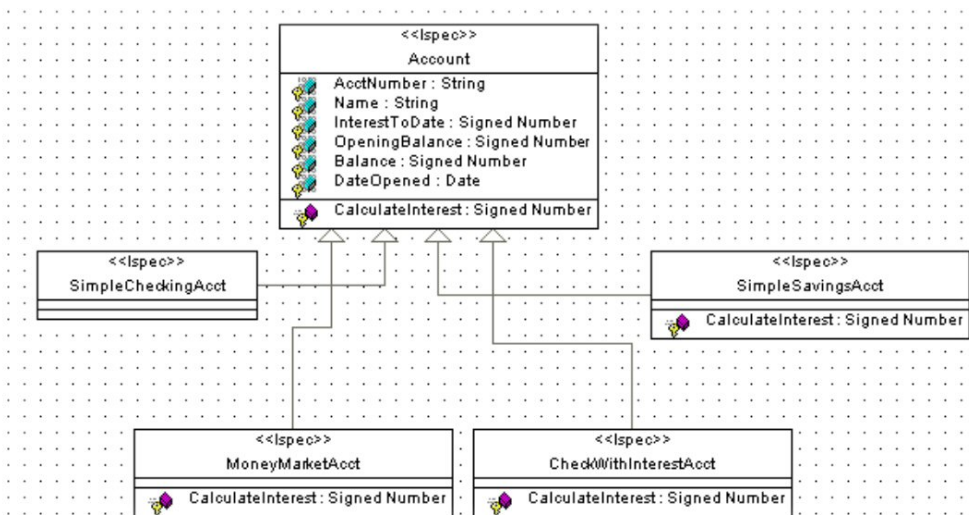
Let's start by looking at how it's done with a more traditional OO tool, such as Java or C#. Here, it is easy enough to model the application class structure to define subclasses that inherit from their parents and override methods, as shown in the Accounts example above. However when it comes to the database design and implementation to support

various account types, that work is usually left to a DBA. The typical approach is to define a separate table for each account type or, perhaps, a single table for all accounts with a separate database structure that holds the account type.

Unfortunately, this makes the programmer's job harder because complex logic is required to access the resulting database. In addition, the developer needs to be very aware of how the database has been implemented. Finally, once these database design decisions have been performed (e.g. to define multiple tables or to define one table with multiple variant record types) and the code has been written, there's little flexibility to make modifications to that implementation. In fact, anything more than the most trivial changes could require a lot of rework.

AB Suite takes a very different approach. Because it generates the entire application including the database, there's no need to involve a DBA. And in the case of polymorphism, all records that come from classes that are derived from an inheritance tree, such as all the types of accounts in our example above, are treated as if they are part of the same database table (when developers want to deal with all types of accounts at the same time) or as if they are stored in their own discrete tables (when developers want to process just the records of a single subclass). And, all of this happens automatically.

To explain, let's return to the Account example. The diagram below shows a simple model of the structure. In this case, the parent class is Account. There are four classes that inherit from Account: SimpleCheckingAcct, MoneyMarketAcct, CheckWithInterestAcct, and SimpleSavingsAcct. Note that in all of the subclasses except SimpleCheckingAcct, the CalculateInterest method has been overridden. That's because the interest calculation process is different for these account types.



Using AB Suite, not only is the application object structure built from this model, as would be the case with languages like Java or C#, but it is also used to produce the database structure. For the AB Suite developer, this results in a much simpler application structure that is easier to write and understand.

For example, when developers write a ForEach loop to retrieve records from the database for the MoneyMarketAcct object, they will get exactly what they expect – all Money Market account records and nothing more.

Similarly, a ForEach over the SimpleCheckingAcct class will return the data associated with Simple Checking accounts.

Now, here's the best part! When developers write a ForEach over the Account class with a special modifier called **Polymorphic**, they will get *all of the records from all account classes*, just as if they were all stored in the same data structure. Not only will the records be retrieved in the proper order, but the LDL+ logic will also automatically apply the correct processing for every subclass it handles. The Account instance variable knows what type of object it is holding, so the correct method will be called automatically.

Inside the ForEach loop, a statement as shown below invokes the correct version of the CalculateInterest method for the type of account it is processing:

```
InterestAmount := ThisAccount.CalculateInterest()
```

What's happening behind the scenes to make this possible? Well, within the AB Suite Framework, an identifier is automatically added to each record to identify the class to which it belongs. When reading through the table polymorphically, each instance is appropriately recognized and processed. This approach simplifies the code and also may reduce data access overhead and improve system performance.

Moreover, if there's a requirement to generate this same model for a different platform that uses a vastly different database management system, there's absolutely no need to make any changes to the application model to handle the new environment. Now *that's* being agile!

## Powerful Polymorphism

Like so many OO terms, polymorphism is new to EAE developers. We hope this article has helped demystify the concept and explain its value. If you're ready to take the next step, you may want to download two whitepapers that provide more about OO topics:

- *Object-Oriented Concepts for EAE Users* by Colin Zealley located in the [eCommunity](#)
- *Agile Business Suite Using Object Orientation in System Modeler* by David Cornell, located on the [Unisys Support web site](#)

And as always, if you have questions about this or any other article in Developing Agility, please email [absuite@unisys.com](mailto:absuite@unisys.com)

## Learning a Business Perspective

### AB Suite Goes to University

Founded in 1919, the University of Latvia in Riga is the largest university in the country. And since the fall of 2006, the 1,700-plus students in the school's Faculty of Mathematics and Computer Science have had the unique opportunity to study business application development in an elective course entitled "AB Suite Programming Environment." Through lecture and hands-on labs, students learn how to solve business problems using the AB Suite development toolset.

"It's common practice at universities in Latvia to team with major technology companies," explains Ed Lacis, vice president of business development, [Baltic Technology Group](#) (BTG). "The goal is to bring real-world experiences and global best practices into the classroom."

Lacis adds, "The schools do a great job when it comes to presenting IT concepts and theory. But they also recognize the value of exposing students to organizations with years of practical technology experience solving day-to-day business challenges for international companies."

A well-regarded IT service provider, BTG was founded in 1991 to bring the talents of Latvia's highly skilled computer professionals to the international market. The company specializes in application development services using Unisys Enterprise Application Environment (EAE), as well as AB Suite, and provides the instructor for the class at the University of Latvia.

### AB Suite Earns an "A"

Student feedback, so far, has been enthusiastic. "Most IT students are already familiar with Visual Studio, so they pick up AB Suite very quickly," observes Lacis. "We originally taught the course with EAE (starting in the spring of 2005). Students responded well but there was a learning curve. Now with AB Suite, they come up to speed quickly and have more time to focus on learning the ins and outs of business application development."

### New Opportunities

The course has been a win-win-win for students, BTG, and Unisys. University of Latvia students gain valuable skills in business application development and, potentially, a career path with BTG. BTG has the advantage of identifying prospective candidates for employment early in the hiring cycle, as well as raising its overall visibility within the new college graduate community. And, Unisys benefits from a growing pool of AB Suite trained talent.

The buzz about Unisys innovative application development tools continues to grow. As of September 2007, the [University of Daugavpils is including the AB Suite Programming Environment course in the core curriculum within the Department of Computer Sciences and Information Technology](#). The department's teaching philosophy is to expose all IT students to *several* real world IT toolsets, and now AB Suite is one of them.

Unisys is putting together an AB Suite Developer certification program! Look for more details, including the proposed certification criteria, in the next issue of Developing Agility.



## **Jump Start Workshops Continue into 2008**

Just how different is Agile Business Suite (AB Suite) from EAE? How easily will our EAE model transfer to AB Suite? What does the run time environment look like? How do I develop a sound migration plan?

We answered these questions – and many others – for the more than 50 students from 18 companies that attended one of our Agile Business Suite (AB Suite) Jump Start educational workshops in North America in 2007.

Starting in February, Unisys and G-Force Global Technologies jointly conducted six of these two-day, hands-on, highly interactive sessions. They are designed to help organizations gain a good understanding of the new AB Suite environment and prepare for migration.

Targeting EAE developers and application development managers, the workshops provide in-depth training on topics ranging from “AB Suite to EAE Mapping” to “Loading an EAE 3.3 MDL file into AB Suite” to “Composite Development.”

The feedback on the workshops has been overwhelmingly positive with an average overall rating of 2.87 on a scale where 3 was the highest rating. Attendees shared the following comments:

- “I knew nothing about AB Suite. Now I actually see the benefits of moving to it.”
- “Learning to navigate within the new product and seeing items that happen in real life are great tips.”
- “It was all new to me and after I move past the shock of Visual Studio, I think I'll like it much better.”
- “I was surprised at the ease of use once one gets accustomed to navigation in Visual Studio. At first glance, it looked more complicated.”
- “Hands-on experience increases my ability to retain training over the long term.”

Special thanks to George McGowan of McGowan Computer Associates and Alan Hood from Unisys for leading the workshops. Another big “thank you” goes to Dick Mullins of Unisys Desktop Services, who worked with Alan to load the workshop PCs with the required EAE, AB Suite, and Microsoft software. These systems traveled to all parts of the United States and Canada this year!

Unisys and G-Force currently are working on plans for more workshops in 2008 – and we strongly encourage all Developing *Agility* readers to attend, if possible. To learn more, check the eCommunity for:

- [General information and schedule](#)
- [A detailed agenda](#)

Questions? Please don't hesitate to contact us at [ABSuite@unisys.com](mailto:ABSuite@unisys.com).

## Annual User Conferences Attract 100s

September and October were busy months for Agile Business Suite (AB Suite) and EAE customers with the UNITE conference taking place in Valley Forge, PA, Future Matters in Bruges, Belgium, and two AB Suite Seminars in St. Paul de Vence, France.

As always, these meetings provided an excellent opportunity for people to learn new things and network with their peers. Read on for more information about each event.

### UNITE 2007

The EAE/AB Suite conference track was one of the most popular at this year's UNITE conference, with an average of 23 attendees per session. As in years past, the program included a mixture of presentations and hands-on labs on a range of topics.

Diane McGonigle's Migration Experiences presentation was particularly well attended. Special thanks go out to Ina Boeke from United Fire & Casualty, Howard Kemple from Nature's Sunshine Products, and George McGowan from McGowan Computer Associates for generously sharing their experiences with AB Suite and taking questions from the audience.

Paul Bourke from the Australian Centre for Unisys Software (ACUS) conducted a two-part lab on the AB Suite Debugger, as well as provided a preview into future directions of AB Suite with Visual Studio Team System/Team Foundation Server. Paul, a leading expert in Unisys on change control, also presented Version Control with AB Suite.

UNITE presentations that were given by Unisys personnel are posted on the [eCommunity](#). Next year's [Annual Technology Conference](#) is scheduled for October 19-23, 2008 at the Caribe Royale Orlando, Florida.

### Future Matters Bruges

Future Matters in Bruges, Belgium devoted nearly an entire day of its two-day schedule to AB Suite presentations. Four ACUS engineers conducted sessions on a broad range of topics, including:

- AB Suite Model Concepts, Grant McCauley
- Debugger Fundamentals, Paul Bourke
- Client Tools, Robert Irvine
- Version Control, Fred Heida

More than 60 people attended the EAE/AB Suite track – an excellent turnout for what has become a very well regarded annual event for Unisys European customers.

The calendar for Future Matters 2008 is still being prepared. Please check the web site for more information about venues and agendas: [www.unisysfuturematters.com](http://www.unisysfuturematters.com)

## Two AB Suite Seminars in St. Paul de Vence

The nearly 150 people who attended the AB Suite Seminars in St. Paul de Vence, France, left the meeting with newfound resolve to migrate to AB Suite. We welcomed attendees from 15 countries this year – and added the convenience of simultaneous translation into French and Italian during the second session.

Inspired by several ACUS presentations, as well as the real-world experiences and lessons-learned from Isle of Man and Post & Co., attendees reported that the meeting was time very well spent. Of course, the meeting location in the south of France was a nice benefit, too!

One particular area of interest was client tools and improving the end-user experience by updating “green” or character-based screens with new graphical user interfaces (GUIs). Attendees were pleased to learn more about how to refresh their existing EAE applications using Component Enabler and new capabilities such as ASP .NET and now AJAX. Once built, these new clients can easily be moved forward into AB Suite, too.

Thanks so much to everyone – from Unisys, ACUS, and our community of EAE and AB Suite customers – who invested their time and talents to making the sessions a resounding success. Special thanks go to LNV Samy, managing director of ACUS, and the five members of his engineering team, who made the long trip from Australia to lead a number of informative sessions.

The 2008 AB Suite Seminars are tentatively scheduled for September 22 – 24 and September 24 – 26. We encourage all European customers to mark their calendars and join us at the [Unisys International Management Centre](#) in gorgeous St. Paul de Vence!

## Calendar

There are many exciting activities coming in 2008, as well as opportunities to meet with fellow Agile Business Suite and EAE users and Unisys personnel around the world. Be sure to check the [eCommunity](#) for the latest information.

What	Where	When
Unisys Technology Forum 2008	Queenstown, New Zealand	May 6-9, 2008
<a href="#">AB Suite Jumpstart Workshops</a>	Locations to be announced in 2008	Dates to be announced in 2008
Agile Business Suite and EAE User Meetings (Choice of two dates)	St. Paul de Vence, France	September 22-24, 2008 September 24-26, 2008
<a href="#">UNITE</a> Annual Technology Conference	<a href="#">Caribe Royale Orlando</a> , Orlando, FL	October 19-23, 2008

© 2007 Unisys Corporation. All rights reserved.

Unisys is a registered trademark of Unisys Corporation. All other brands and products referenced herein are acknowledged to be trademarks or registered trademarks of their respective holders.

12/07