



Table des matières

Questions-réponses client : Migration de la société DE CLARENS vers AB Suite

Une société de courtage en assurances vieille d'un siècle passe à AB Suite dans un environnement de production Windows. M. René Conte, directeur informatique, nous explique comment DE CLARENS a planifié ce changement.

Page 2

Le coin des ingénieurs : Polymorphisme

Le polymorphisme aide les développeurs à faire ce qu'ils préfèrent : écrire du code propre et élégant, facile à gérer.

Page 4

Apprendre avec une perspective d'entreprise

Au programme dans deux universités, AB Suite aide les étudiants de l'ancienne Union Soviétique à appliquer les pratiques de développement occidentales pour résoudre les défis mondiaux des entreprises.

Page 8

Les ateliers AB Suite Jump Start (pieds à l'étriee) continuent en 2008

Plus de 50 étudiants de 18 sociétés ont participé aux sessions en 2007. Découvrez ce que vous pouvez apprendre en y participant.

Page 9

Les conférences annuelles pour utilisateurs ont attiré plus de 100 personnes

Les séminaires UNITE, Future Matters et AB Suite ont proposé d'excellentes opportunités en matière de formation et de mise en réseau au deuxième semestre 2007.

Page 10

Developing Agility en français? Oui !

Une traduction [française](#) de Developing Agility est désormais disponible. Ce numéro, en est la deuxième version, rejoignant ainsi les versions [espagnole](#) et [portugaise](#).

Support technique Windows Vista

À partir de novembre 2007, grâce à IC 3220, EAE Developer 3.0 est pris en charge sur Windows Vista. AB Suite Developer 1.2 est pris en charge par Windows Vista depuis septembre 2007 grâce à IC 1350. Consultez le [site Web d'assistance des produits Unisys](#) pour plus d'informations.

Événements

Consultez notre calendrier pour connaître les dernières infos sur les plans 2008.

Page 11

Questions-réponses client : Migration de la société DE CLARENS vers AB Suite

Rubrique régulière de notre lettre d'information Developing Agility, les Questions-réponses client exposent des situations du monde réel et des conseils pratiques concernant la migration et l'utilisation d'Agile Business Suite (AB Suite).

Ce numéro propose une interview de M. René Conte, directeur informatique chez DE CLARENS, une société de courtage en assurances vieille d'un siècle dont le siège social se trouve à Paris, en France. Comme beaucoup d'entreprises, DE CLARENS dépend de son infrastructure informatique et de ses capacités de développement d'applications pour assurer la croissance de ses différentes offres de produits et services. Depuis plus de 30 ans, DE CLARENS travaille avec Unisys pour ses solutions technologiques et ses services de consulting. Notre discussion avec M. Conte décrit comment une récente migration vers AB Suite leur a permis d'assurer une croissance commerciale continue.

Developing Agility : Voulez-vous bien nous décrire le principal enjeu commercial de DE CLARENS et nous expliquer comment l'informatique a résolu ce défi ?

René Conte : DE CLARENS cherche à faire croître son entreprise, ce qui peut impliquer la fusion ou l'acquisition d'autres courtiers. Notre direction comprend que des investissements dans le domaine informatique peuvent favoriser cette expansion, spécialement les nouvelles capacités d'intégration. Le département informatique devait trouver de nouveaux moyens pour relier rapidement les applications des nouveaux partenaires à nos applications.

Developing Agility : Parlez-nous de l'environnement informatique chez DE CLARENS.

René Conte : Notre application coeur de métier, SARA, est une solution personnalisée que nous avons développée pour suivre les clients, les contrats, les garanties, etc. Au départ, elle était écrite en COBOL, puis nous l'avons réécrite en EAE avant de migrer maintenant vers AB Suite. SARA peut accueillir 500 Reports et Specs et est hébergée sur une plate-forme Microsoft Windows. Nous disposons également de nombreuses autres applications du marché, par exemple pour la paie, qui ne sont pas écrites en L4G, mais qui sont interfacées avec SARA.

DE CLARENS emploie deux développeurs à temps plein et compte sur des ressources externes pour compléter ses effectifs, selon les besoins. Nous prenons en charge plus de 100 utilisateurs finaux.

Avant de passer à AB Suite, nous avons modernisé nos interfaces utilisateur final grâce à un générateur ASP personnalisé développé par Unisys France.

« C'était très important de conserver notre application EAE. ... et nous souhaitions également une solution avec d'excellentes possibilités d'intégration. »

Developing Agility : Décrivez-nous votre processus de prise de décision lors du choix de passer au nouvel environnement de développement dans AB Suite.

René Conte : C'était très important de conserver notre application EAE. Et comme je l'ai mentionné précédemment, nous souhaitions également une solution avec d'excellentes possibilités d'intégration. EAE propose quelques fonctionnalités, mais AB Suite permet une intégration vraiment très simple dans Visual Studio. Par exemple, nous pouvons appeler le code C++ et C# directement à partir de SARA grâce à AB Suite.

Developing Agility : Quels avantages métiers pensez-vous tirer d'AB Suite pour votre entreprise ?

René Conte : Avec AB Suite, DE CLARENS peut passer à un environnement de développement moderne et préserver l'investissement réalisé dans notre solution EAE. En outre, nous pouvons l'intégrer très facilement et rapidement avec des services existant dans d'autres applications écrites avec d'autres langages. Nos directeurs demandent toujours de nouvelles interfaces utilisateurs graphiques et un meilleur accès aux applications à partir du Web. Désormais, grâce au nouveau générateur d'écran d'AB Suite et au générateur ASP personnalisé, nous pouvons offrir rapidement ces fonctionnalités, et avec moins d'efforts.

Developing Agility : Quelles sont les fonctionnalités d'AB Suite qui ont eu une grande influence lors de votre choix ?

René Conte : Tout d'abord, je craignais de perdre des données et du temps de développeurs. Donc, le fait que le passage d'EAE à AB Suite soit sans risque était extrêmement important. Ensuite, les fonctionnalités OO furent déterminantes également, car elles nous permettaient de répondre rapidement aux requêtes de la direction de DE CLARENS. Finalement, je dois souligner l'importance de l'intégration avec d'autres applications.

« La migration fut très simple, y compris le passage du modèle de données actuel vers AB Suite. »

Developing Agility : Pouvez-vous raconter à nos lecteurs comment DE CLARENS a planifié et mené sa migration ?

René Conte : Comme vous le savez, DE CLARENS travaille avec Unisys depuis 1975. Il y a cinq ans, Unisys et DE CLARENS ont élaboré un plan de route technologique que nous avons suivi depuis. Nous travaillons en étroite collaboration avec les consultants Unisys sur la migration vers AB Suite. Les consultants Unisys ont d'abord converti notre fichier source applicatif EAE vers AB Suite dans les locaux d'Unisys. Ensuite, ils ont installé AB Suite sur notre site, ont chargé le nouveau modèle applicatif et nous ont laissé la solution migrée pour un mois à l'essai. Nous avons prévu de mettre tout en production début 2008.

Developing Agility : La migration fut-elle facile ? Qu'avez-vous appris lors du processus ?

René Conte : La migration fut très simple, y compris le passage du modèle applicatif vers AB Suite. Nous n'avons rien dû faire de spécial et les résultats étaient très propres. Cette étape a donc été très simple. Nous avons appris que nous pouvions migrer rapidement au moindre coût.

Developing Agility : Parlez-vous de votre expérience d'utilisation d'AB Suite dans un environnement de production.

René Conte : Nous avons prévu de mettre l'application AB Suite en production au début de l'année 2008, mais pendant notre période de test, nous n'avons constaté aucune différence de fonctionnement. Nos utilisateurs finals n'ont rencontré aucun changement et bénéficient presque de la même efficacité et des mêmes performances. Les ASP exécutées avec EAE fonctionnent aussi bien dans AB Suite, sans aucun changement.

Developing Agility : Que pensent vos développeurs du nouvel outil AB Suite ?

René Conte : Nos développeurs ont suivi la formation d'Unisys sur la migration et les différences entre EAE et AB Suite. Comme vous le savez, les concepts d'EAE et d'AB Suite sont fort différents. Avec EAE, nous utilisons des Reports et des Specs. Avec AB Suite, nous avons des méthodes et des classes. C'est une vraie différence de mentalité. Maintenant, nos développeurs utilisent AB Suite comme ils utilisaient EAE. Je veux dire par là qu'ils n'utilisent pas tous

les avantages des fonctionnalités OO d'AB Suite, enfin, pas encore. Mais cela va changer. Nous avons conclu un contrat d'assistance avec Unisys afin que leurs consultants travaillent côte à côte avec nos développeurs, de manière informelle, pendant quelques semaines, afin de leur apprendre à utiliser AB Suite correctement.

Developing Agility : Quels sont vos plans concernant AB Suite dans le futur ?

René Conte : Nous voulons apprendre à travailler « à la manière d'AB Suite ». DE CLARENS a d'autres plans pour faire croître la société, donc nous aurons de nouvelles solutions à intégrer. Tout nouveau développement se fera avec AB Suite.

Developing Agility : Quels conseils donneriez-vous aux autres entreprises qui migrent vers AB Suite ?

René Conte : Je recommande vraiment AB Suite à toutes les entreprises qui utilisent EAE, surtout à celles qui doivent intégrer d'autres solutions. AB Suite facilite la création d'applications composites. Je pense également qu'il faut demander l'assistance des consultants Unisys, car leur aide facilite grandement la migration.

Merci pour cet entretien agréable, René ! Nous vous souhaitons à vous et votre équipe tout ce qu'il y a de mieux à l'avenir avec AB Suite. Pour plus d'informations sur DE CLARENS, veuillez consulter le site www.declarens.com.

Le coin des ingénieurs : Polymorphisme

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

Dans Le coin des ingénieurs, nous traitons des concepts de l'OO, tels que les stéréotypes ([2Q2007](#)) et l'encapsulation ([3Q2007](#)). De quoi allons-nous parler maintenant ? Du polymorphisme, évidemment ! Oui, encore un autre terme long et inconnu, mais auquel les développeurs doivent s'intéresser, car il offre de nombreux avantages. Le polymorphisme (tout comme l'héritage, l'encapsulation et les autres constructions OO) aide les développeurs à faire ce qu'ils préfèrent : écrire du code très propre et élégant, extrêmement facile à gérer.

Qu'est-ce que le polymorphisme ?

Le polymorphisme est un terme couramment utilisé en biologie. Si vous faites une recherche sur Google, vous trouverez sans doute une définition telle que celle du American Heritage Science Dictionary :

polymorphisme (polimɔrfism) Existence de deux ou plusieurs formes dans un organisme adulte de la même espèce. Chez les abeilles, la présence d'une reine, d'ouvrières et de faux bourdons est un exemple de polymorphisme.

Dans OO, le concept est semblable et, comme pour les abeilles, mieux compris à travers un exemple. Les banques utilisent différents types de comptes : compte chèques, compte d'épargne ordinaire, compte à terme, plan d'épargne, comptes titres... Et la liste ne cesse de s'allonger, car les institutions financières cherchent à se différencier sur le marché.

Fondamentalement, il s'agit de types de comptes différents qui partagent tous certaines caractéristiques telles que le numéro de compte et le solde. Ces attributs sont définis une fois au niveau de la classe de compte et chaque sous-classe en hérite. Évidemment, chaque sous-classe a ses propres attributs et comportements, mais l'idée est de

favoriser une généralité autant que possible afin de simplifier le développement, d'améliorer la réutilisabilité et de diminuer la maintenance.

Il est vrai que ces comptes rapportent des intérêts de différentes manières. Les comptes chèques rapportent peu ou pas d'intérêt tandis que les autres en rapportent à des taux différents. Une des approches traditionnelles pour programmer ces règles consiste à définir un sous-programme qui calculera l'intérêt pour chaque type de compte en fonction de paramètres tels que le type de compte, les dates et les valeurs financières.

Avec les concepts OO, une autre approche sera possible en définissant le calcul de l'intérêt comme une méthode au sein de chaque type de compte. Par exemple, chaque type de compte pourrait posséder sa propre méthode CalculerIntérêt. Cependant, tous les comptes ont en commun le calcul d'intérêt, même si c'est « aucun calcul ». Par conséquent, grâce au polymorphisme, la meilleure approche est de définir la méthode CalculerIntérêt au niveau du Compte et de laisser le choix réel de la manière de réaliser le calcul au moment de l'exécution en fonction du type de compte.

Prenons l'exemple suivant. Pour commencer, la méthode « standard » CalculerIntérêt est définie dans la classe Compte de base. Ensuite, pour les sous-classes différentes, on définit une variante locale de la méthode CalculerIntérêt, qui *écrase* la méthode de la classe supérieure (Compte).

C'est possible grâce au polymorphisme, et voilà la syntaxe obtenue :

```
MontantIntérêt := CeCompte.CalculerIntérêt()
```

Avec le polymorphisme, l'instruction ci-dessus n'est écrite qu'une seule fois et appliquée à tous les types de comptes. Au moment de l'exécution, le système peut déterminer le type de classe affecté à la variable locale CeCompte. Par conséquent, lorsque l'instruction est exécutée sur un compte Épargne, c'est la version Épargne de la méthode CalculerIntérêt qui est appelée. Lorsque la même instruction est exécutée sur un compte chèques, c'est alors la version CompteChèques de CalculerIntérêt qui est appelée.

Le polymorphisme supprime le besoin d'une logique complexe telle que les instructions Case ou If imbriquées, pour obtenir le bon résultat. À la place, la méthode appropriée est *automatiquement* appelée pour le type de compte traité.

Dans cet exemple, le polymorphisme offre d'importantes possibilités aux développeurs de l'application bancaire. Tout d'abord, ils peuvent écrire une méthode de calcul de l'intérêt commune à la plupart des types de comptes, méthode qui sera automatiquement héritée et utilisée par toutes les sous-classes. Cependant, pour les comptes dont le calcul d'intérêt est différent ou plus complexe, les développeurs peuvent écrire une méthode portant le même nom spécialement pour cette sous-classe et faire écraser la méthode à partir de la classe supérieure Compte. Donc, si la banque décide de proposer un nouveau compte avec un taux d'intérêt spécial pour les jeunes adultes de 21 à 29 ans, il ne faut pas changer la boucle PourChaque ou l'instruction qui appelle CalculerIntérêt. Il suffit de créer une nouvelle sous-classe de Compte avec son propre calcul d'intérêt et c'est fait !

En coulisses : conception de la base de données pour prendre en charge le polymorphisme

Un des aspects intéressants du polymorphisme dans AB Suite, c'est l'implémentation physique au niveau de la base de données.

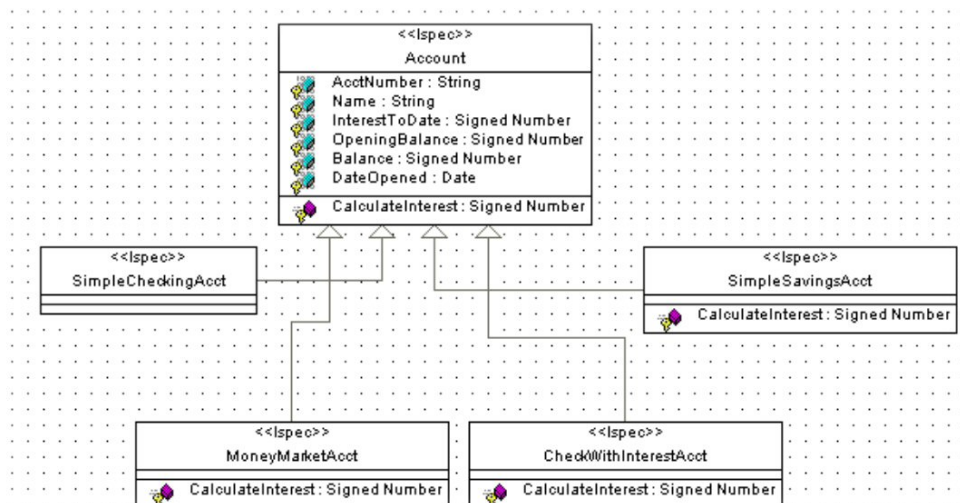
Examinons d'abord comment cela se passe avec un outil OO plus classique, par exemple, Java ou C#. Ici, il est assez facile de modéliser la structure de la classe applicative pour définir des sous-classes qui héritent de leurs parents et

écrasent les méthodes, comme illustré dans l'exemple Comptes ci-dessus. Cependant, quand on en arrive à la conception et l'implémentation de la base de données pour prendre en charge différents types de comptes, le travail est généralement laissé à un ABD (administrateur de bases de données). L'approche classique consiste à définir une table séparée pour chaque type de compte ou, éventuellement, une seule table pour tous les comptes avec une structure de base de données séparée qui contient le type de compte.

Malheureusement, cela complique la tâche du programmeur, car il faut une logique complexe pour accéder à la base de données qui en résulte. En outre, le développeur doit bien savoir comment la base de données a été implémentée. Enfin, lorsque les décisions en matière de conception de la base de données ont été prises (c-à-d définir les différentes tables ou une table avec plusieurs types d'enregistrements) et que le code a été écrit, il existe peu de possibilités pour apporter des modifications à cette implémentation. En réalité, à part pour de petits changements, les modifications demandent beaucoup de travail.

AB Suite propose une approche fort différente. Étant donné qu'AB Suite crée toute l'application, y compris la base de données, il n'est pas nécessaire d'impliquer un ABD. Dans le cas du polymorphisme, tous les enregistrements en provenance des classes dérivées d'une arborescence en héritage, tels que tous les types de comptes dans notre exemple précédent, sont traités comme s'ils faisaient partie de la même table de base de données (si les développeurs veulent traiter tous les types de comptes en même temps) ou comme s'ils étaient enregistrés dans leurs propres tables (si les développeurs veulent traiter simplement les enregistrements d'une seule sous-classe). Et tout ça se fait automatiquement.

Pour l'expliquer, reprenons l'exemple des comptes bancaires. Le diagramme ci-dessous illustre un modèle simple de structure. Dans ce cas, la classe parent est Compte (Account). Quatre classes héritent de la classe Compte : CpteChèque (SimpleCheckingAcct), CpteTitre (MoneyMarketAcct), ChèqueRémunéré (CheckWithInterestAcct) et CpteÉpargneSimple (SimpleSavingsAcct). Notez que dans toutes les sous-classes, excepté CpteChèque, la méthode CalculerIntérêt a été écrasée. C'est parce que le processus de calcul d'intérêt est différent pour ces types de comptes.



Avec AB Suite, la structure de l'objet applicatif n'est pas seulement créée à partir de ce modèle, comme ce serait le cas avec les langages Java ou C#, mais elle est également utilisée pour produire la structure de la base de données. Cela permet au développeur d'AB Suite de bénéficier d'une structure d'application beaucoup plus simple à écrire et à comprendre.

Par exemple, lorsque les développeurs écrivent une boucle PourChaque pour récupérer les enregistrements de la base de données pour l'objet CpteTitre, ils recevront exactement ce qu'ils attendent : tous les enregistrements des comptes Marché monétaire et rien d'autre.

De même, une boucle PourChaque sur une classe CpteChèque renverra les données associées aux comptes chèques.

Et maintenant, voilà le meilleur ! Lorsque les développeurs écrivent une boucle PourChaque sur la classe Compte avec un modificateur spécial appelé **Polymorphic (polymorphique)**, ils recevront *tous les enregistrements de toutes les classes de compte*, comme s'ils étaient enregistrés dans la même structure de données. Non seulement les enregistrements seront récupérés dans le bon ordre, mais encore la logique LDL+ appliquera automatiquement le traitement correct pour chaque sous-classe. La variable de l'instance Compte sait quel type d'objet elle contient, donc la méthode correcte sera automatiquement appelée.

Dans chaque boucle PourChaque, une instruction telle que ci-dessous appelle la version correcte de la méthode CalculerIntérêt pour le type de compte traité.

```
MontantIntérêt := CeCompte.CalculerIntérêt()
```

Que se passe-t-il dans les coulisses pour que cela soit possible ? Et bien, dans le cadre AB Suite, un identifiant est automatiquement ajouté à chaque enregistrement afin d'identifier la classe à laquelle il appartient. Lorsque vous lisez la table de manière polymorphique, chaque instance est correctement reconnue et traitée. Cette approche simplifie le code, peut diminuer les accès inutiles aux données et améliore les performances du système.

De plus, s'il faut générer ce même modèle pour une autre plate-forme qui utilise un système de gestion de base de données différent, il ne sera pas nécessaire de modifier la source de l'application pour gérer le nouvel environnement. C'est ça l'agilité !

Puissant polymorphisme

Comme beaucoup de termes OO, le polymorphisme est nouveau pour les développeurs EAE. Nous espérons que cet article vous a aidé à démystifier le concept et à en expliquer les avantages. Si vous êtes prêt à passer à l'étape suivante, vous pouvez télécharger deux documents (en anglais) qui vous en diront plus sur les rubriques OO :

- *Object-Oriented Concepts for EAE Users* par Colin Zealley, à la page [eCommunity](#)
- *Agile Business Suite Using Object Orientation in System Modeler* par David Cornell, sur le [site Web d'assistance des produits Unisys](#)

Et comme toujours, si vous avez des questions à propos d'un article paru dans Developing Agility, vous pouvez envoyer un message électronique à l'adresse suivante absuite@unisys.com.

Apprendre avec une perspective d'entreprise

AB Suite va à l'université

Fondée en 1919, l'Université de Lettonie, à Riga, est la plus grande université du pays. Et depuis l'automne 2006, les plus de 1 700 étudiants de la Faculté de mathématique et d'informatique de l'école ont l'opportunité unique d'étudier le développement d'application en entreprise dans un cours intitulé « Environnement de programmation AB Suite ». À travers conférences et travaux pratiques, les étudiants apprennent à résoudre des problèmes d'entreprise à l'aide de l'outil de développement AB Suite.

« Il est fréquent, dans les universités de Lettonie, de s'associer avec des sociétés technologiques importantes », explique Ed Lacis, vice-président du développement de l'entreprise, chez [Baltic Technology Group](#) (BTG).
« L'objectif est d'amener dans la salle de classe des expériences du monde réel et les meilleures pratiques. »

Lacis ajoute : « Les écoles font du bon travail quand il s'agit de présenter des concepts et théories informatiques. Mais elles reconnaissent aussi l'utilité d'exposer les étudiants à des organisations jouissant de plusieurs années d'expérience technologique pour résoudre les défis quotidiens auxquels sont confrontées les entreprises internationales. »

Fournisseur de services informatiques bien considéré, BTG fut fondé en 1991 pour porter sur le marché international les talents des professionnels lettons en informatique hautement qualifiés. La société se spécialise dans les services de développement d'application avec Unisys Enterprise Application Environment (EAE) et AB Suite, et fournit un formateur pour les cours à l'Université de Lettonie.

AB Suite a obtenu la mention « très bien »

Jusqu'à présent, les appréciations des étudiants sont très enthousiastes. « La plupart des étudiants en informatique connaissent déjà Visual Studio, donc ils accrochent très rapidement avec AB Suite », observe Lacis. « Au départ, nous avons enseigné le cours avec EAE (au printemps 2005). Les étudiants ont bien répondu, mais il y avait une courbe d'apprentissage. Maintenant, avec AB Suite, ils progressent rapidement et ont plus de temps pour se consacrer à l'apprentissage de tous les tenants et aboutissants du développement d'application d'entreprise. »

Nouvelles opportunités

Le cours fut un triple succès : pour les étudiants, BTG et Unisys. Les étudiants de l'Université de Lettonie ont fortement amélioré leurs compétences en développement d'application d'entreprise et potentiellement gagné une opportunité de carrière avec BTG. BTG a l'avantage d'identifier des candidats éventuels à un emploi tôt dans le cycle d'embauche et d'améliorer sa visibilité générale au sein des nouveaux diplômés. Et Unisys profite d'un groupe de plus en plus important de talents formés à AB Suite.

Le foisonnement autour des outils de développement d'application innovants d'Unisys ne fait qu'augmenter. Depuis septembre 2007, [l'Université de Daugavpils inclut le cours Environnement de programmation AB Suite dans son programme principal pour le Département des sciences informatiques et des technologies de l'information](#). La philosophie pédagogique du département est d'exposer tous les étudiants en technologie de l'information à *plusieurs* outils informatiques du monde réel, et AB Suite en fait partie.

Unisys met sur pied un programme de certification Développeur AB Suite ! Vous trouverez plus de détails, dont les critères de certification proposés, dans le prochain numéro de Developing Agility.

Les ateliers AB Suite Jump Start (pieds à l'étrée) continuent en 2008

En quoi AB Suite est-il différent d'EAE ? Avec quelle facilité votre modèle EAE passera-t-il vers AB Suite ? À quoi ressemble l'environnement d'exécution ? Comment puis-je développer un plan de migration sûr ?

Nous avons répondu à ces questions, et à beaucoup d'autres, aux plus de 50 étudiants de 18 entreprises qui ont participé à l'un de nos ateliers de formation AB Suite Jump Start en Amérique du Nord en 2007.

Depuis février, Unisys et G-Force Global Technologies ont organisé conjointement six de ces sessions de deux jours, interactives et pratiques. Elles ont pour but d'aider les entreprises à mieux comprendre le nouvel environnement AB Suite et à se préparer à la migration.

Destinés particulièrement aux développeurs EAE et aux responsables du développement d'application, les ateliers offrent une formation détaillée sur des thèmes allant de « Mapping AB Suite-EAE » à « Chargement d'un fichier EAE 3.3 MDL dans AB Suite », en passant par « Développement composite d'applications ».

Les réactions suite aux ateliers furent très positives. Ils ont été évalués à 2.87 sur une échelle allant jusqu'à 3. Les participants ont fait partager les commentaires suivants :

- « Je ne connaissais rien à AB Suite. Maintenant, je mesure réellement les avantages d'y passer. »
- « Apprendre à naviguer dans le nouveau produit et voir les éléments se dérouler dans la vie réelle, c'est très utile. »
- « « C'était tout nouveau pour moi et quand j'aurai dépassé le choc de Visual Studio, je crois que ça me plaira beaucoup. »
- « J'ai été surpris par la facilité d'utilisation quand on est habitué à la navigation dans Visual Studio. À première vue, ça semblait plus compliqué. »
- « L'expérience pratique me permet de mieux retenir la formation à long terme. »

Un grand merci à George McGowan de McGowan Computer Associates et à Alan Hood d'Unisys pour avoir mené ces ateliers. Un autre grand merci à Dick Mullins de Unisys Desktop Services, qui a travaillé avec Alan pour installer sur les ordinateurs de l'atelier les logiciels EAE, AB Suite et Microsoft nécessaires. Ces systèmes ont voyagé dans tous les États-Unis et le Canada cette année ! Ces ateliers sont aussi disponibles en Europe. N'hésitez pas à consulter nos consultants AB Suite.

Unisys et G-Force travaillent actuellement sur la préparation d'autres ateliers en 2008. Et nous encourageons tous les lecteurs de Developing Agility à y participer, si possible. Pour en savoir plus, consultez la section eCommunity pour

- [des informations générales et un calendrier](#)
- [un programme détaillé](#)

Des questions ? N'hésitez pas à nous contacter à l'adresse ABSuite@unisys.com.

Les conférences annuelles pour utilisateurs ont attiré plus de 100 personnes

Septembre et octobre furent des mois fort chargés pour les clients AB Suite et EAE. Il y a eu la conférence UNITE qui a eu lieu à Valley Forge en Pennsylvanie, la conférence Future Matters à Bruges en Belgique et deux séminaires AB Suite à St Paul de Vence, en France.

Comme toujours, ces réunions ont permis aux gens d'apprendre de nouvelles choses et de rencontrer leurs collègues. Voici d'autres informations sur chaque événement.

UNITE 2007

La conférence EAE/AB Suite fut l'une des plus prisées lors de la conférence UNITE de cette année. Il y eut une moyenne de 23 participants par session. Comme lors des années précédentes, le programme consistait en différentes présentations et laboratoires pratiques sur une gamme de sujets.

La présentation Expériences de migration, par Diane McGonigle, a particulièrement attiré beaucoup de participants. Merci plus particulièrement à Ina Boeke de United Fire & Casualty, Howard Kemple de Nature's Sunshine Products, et George McGowan de McGowan Computer Associates pour avoir partagé généreusement leurs expériences avec AB Suite et répondu aux questions du public.

Paul Bourke du Australian Centre for Unisys Software (ACUS) a dirigé le laboratoire en deux parties sur AB Suite Debugger ; il a également présenté l'orientation future d'AB Suite avec Visual Studio Team System/Team Foundation Server. Paul, spécialiste chez Unisys dans le contrôle des changements, a aussi présenté Version Control avec AB Suite.

Les présentations UNITE proposées au personnel d'Unisys sont postées sur le site, à la page [eCommunity](#). La [conférence annuelle sur les technologies](#) de l'année prochaine est programmée du 19 au 23 octobre 2008 au Caribe Royale Orlando en Floride.

Future Matters Bruges

Future Matters, à Bruges en Belgique, a consacré presque une journée entière, sur les deux jours du programme, aux présentations d'AB Suite. Quatre ingénieurs de l'ACUS ont dirigé des sessions sur un grand nombre de sujets, dont :

- Concepts du modèle AB Suite, par Grant McCauley
- Principes fondamentaux du débogage, par Paul Bourke
- Outils client, par Robert Irvine
- Version Control, par Fred Heida

Plus de 60 personnes ont participé à la section EAE/AB Suite, un excellent chiffre pour ce qui est devenu un événement annuel fort attendu pour les clients Unisys européens.

Le calendrier de Future Matters 2008 est en cours de préparation. Veuillez consulter le site Web pour plus d'informations sur les événements et programmes : www.unisysfuturematters.com

Deux séminaires AB Suite à St Paul de Vence

Les 150 personnes environ qui ont participé aux séminaires AB Suite à St Paul de Vence, en France, ont quitté les sessions avec de nouvelles bonnes résolutions pour passer à AB Suite. Cette année, nous avons accueilli des participants de 15 pays et avons assuré la traduction simultanée en français et en italien lors de la deuxième session.

Inspirés par différentes présentations d'ACUS ainsi que par des expériences dans le monde réel et les leçons tirées des sociétés Isle of Man et Post & Co, les participants nous ont signalé qu'ils n'avaient pas perdu leur temps à ces séminaires. Évidemment, le lieu du séminaire, le sud de la France, n'y était pas pour rien !

Deux sujets ont particulièrement intéressé les participants : les outils clients et l'amélioration de l'expérience de l'utilisateur final grâce à la modernisation des écrans « verts » basés sur une interface par de nouvelles interfaces utilisateurs graphiques. Les participants furent ravis d'en savoir plus sur la manière d'actualiser leurs applications EAE existantes grâce à Component Enabler et aux nouvelles fonctionnalités telles que ASP .NET et AJAX. Une fois créées, ces nouvelles interfaces clients peuvent facilement être reprises avec AB Suite.

Merci beaucoup à toutes les personnes d'Unisys, de l'ACUS et à tous nos clients d'EAE et AB Suite, qui ont investi leur temps et leur talent pour que ces sessions soient une vraie réussite. Un merci tout spécial à LNV Samy, administrateur délégué d'ACUS, et aux cinq membres de son équipe d'ingénieurs, qui ont fait le long voyage depuis l'Australie pour venir diriger ces sessions d'information.

Les séminaires AB Suite 2008 sont provisoirement programmés du 22 au 24 septembre et du 24 au 26 septembre. Nous encourageons tous les clients européens à bloquer dès à présent les dates dans leur agenda et à nous rejoindre au [Unisys International Management Centre](#) dans le magnifique village de St Paul de Vence !

Calendrier

En 2008, de nombreuses activités intéressantes sont programmées, ainsi que des possibilités de rencontre avec d'autres utilisateurs d'EAE et AB Suite et avec des membres du personnel d'Unisys, dans le monde entier. N'oubliez pas de consulter [eCommunity](#) pour les dernières nouvelles.

Quoi ?	Où ?	Quand ?
Unisys Technology Forum 2008	Queenstown, Nouvelle Zélande	6 au 9 mai 2008
Ateliers AB Suite Jumpstart	Lieu à préciser en 2008	Dates à préciser en 2008
Réunions d'utilisateurs AB Suite et EAE (choix entre deux dates)	St. Paul de Vence, France	22 au 24 septembre 2008 24 au 26 septembre 2008
UNITE Conférence annuelle sur les technologies	Caribe Royale Orlando , Orlando, Floride	19 au 23 octobre 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