



Developing Agility

*Lettre d'information trimestrielle destinée
aux clients EAE et AB Suite d'Unisys*

Table des matières

- Page 2 | **Entretien avec notre client : l'assureur néerlandais Movir**
L'ingénieur système Ron Thakoer nous parle de la migration de son entreprise vers Agile Business Suite.
- Page 4 | **Le coin des ingénieurs : Portez les applications migrées vers de nouvelles sphères grâce à la modélisation dans une optique de réutilisation**
Grant McCauley dévoile quatre moyens de doper la réutilisabilité dans les systèmes existants une fois la migration vers Agile Business Suite effectuée.
- Page 9 | **Silverlight introduit une nouvelle option d'interface utilisateur**
Découvrez dans les grandes lignes comment utiliser cette nouvelle technologie dans une application EAE ou Agile Business Suite.
- Page 12 | **Calendrier**
Consultez notre calendrier pour connaître les dernières informations sur les événements à venir.

Entretien avec notre client : l'assureur néerlandais Movir

L'un des objectifs de *Developing Agility* est de tenir les clients EAE informés de la façon dont certains de leurs homologues ont procédé pour planifier et mettre en œuvre avec succès leur migration vers Agile Business Suite. Cet article, qui s'inscrit dans cette optique, présente les activités et les réalisations de Movir, une compagnie d'assurance spécialisée située aux Pays-Bas qui propose des assurances invalidité aux professionnels indépendants en activité néerlandais, ainsi qu'une couverture santé.

Nous nous sommes récemment entretenus avec Ron Thakoer, ingénieur système chez Movir, au sujet de leur récente migration vers Agile Business Suite dans l'environnement ClearPath MCP. Lisez cet article pour en savoir plus sur le processus de migration de Movir d'EAE vers Agile Business Suite et connaître l'opinion de Ron Thakoer sur les principaux avantages dont il espère tirer profit en tant que nouvel utilisateur d'Agile Business Suite.

Developing Agility (DA) : *Parlez-nous de l'environnement informatique chez Movir. De combien d'applications disposez-vous ? Quelles fonctions métier prennent-elles en charge ? Sont-elles volumineuses ? Et quelle est la taille de votre équipe de développement ?*

Ron Thakoer (RT) : Movir exploite trois applications EAE principales, deux d'entre elles supportent directement les activités de l'entreprise tandis que la troisième est un simple outil de contrôle personnalisé. Notre principale application, MOSYS, qui regroupe 120 ISPEC et 200 REPORTS, comprend une base de données de production d'environ 10 Go et est utilisée par quelque 110 utilisateurs finaux.

La deuxième application que nous utilisons, qui est bien moins volumineuse, est semblable au système MOSYS, excepté qu'elle est spécifiquement conçue pour prendre en charge les services d'assurance destinés aux propriétaires de petites entreprises.

Nous disposons de cinq développeurs Agile Business Suite.

DA : *Quels avantages métier pensez-vous tirer d'Agile Business Suite ?*

RT : Nous avons lancé le nouveau système en juin 2009 et laissé dans un premier temps notre équipe informatique se familiariser avec le nouvel environnement. Une fois cette période de lancement écoulée, nous espérons exploiter Agile Business Suite pour optimiser certains processus, par exemple en renforçant l'intégration entre Agile Business Suite et nos applications de guichet.

De plus, nous avons prévu d'utiliser Agile Business Suite pour étendre le périmètre d'une fonctionnalité que nous avons développée sous Microsoft® Word et qui automatise la production des courriers. Nous exploitons également Infolmage, un système de gestion documentaire d'Unisys, et nous espérons en outre simplifier le processus d'intégration de cet outil dans les applications Agile Business Suite.

DA : *Qu'en est-il des avantages techniques ?*

RT : Bien entendu, nous espérons qu'Agile Business Suite nous permettra d'améliorer la productivité de nos développeurs mais, plus important encore, nous sommes impatients de travailler dans un environnement orienté objet (OO). Forts de notre environnement OO, nous pourrions optimiser la structure de nos applications, ce qui nous permettra de les maintenir et de les améliorer plus rapidement et plus efficacement.

DA : *Avant de migrer vers Agile Business Suite, vous avez migré l'interface graphique de votre application EAE de PowerClient (Graphical Interface Workbench) vers Microsoft Visual Basic® .NET. Pourquoi avoir choisi Visual Basic .NET ? Vos utilisateurs finaux ont-ils remarqué le moindre changement après la mise en œuvre d'Agile Business Suite ?*

RT : Nous utilisons une interface graphique PowerClient et Visual Basic 6 intégrée pour notre application EAE. Cependant, un support de premier niveau devant être assuré finalement pour les deux produits, nous savions qu'il était temps de migrer vers des technologies plus récentes. De plus, étant donné que nous utilisons déjà Visual Basic 6, il nous semblait logique d'un point de vue opérationnel de migrer vers Visual Basic .NET en utilisant le générateur inclus dans les outils client EAE et Agile Business Suite. Nous avons également enrichi l'interface utilisateur final générée d'un code personnalisé au moyen de Visual Basic .NET. >>

Nous avons énormément gagné en efficacité côté utilisateur final. À titre d'exemple, lorsqu'une fonction de Visual Basic .NET est appelée, nos utilisateurs peuvent désormais rester dans la même fenêtre. En commençant par moderniser cette interface, nous avons permis aux utilisateurs de l'utiliser avec les applications EAE pendant quelques mois, avant la migration vers Agile Business Suite. De plus, nous avons pu migrer nos interfaces utilisateur graphiques Visual Basic .NET mises à jour vers Agile Business Suite sans le moindre problème et les utilisateurs n'ont remarqué aucune différence.

DA : *Quelles sont les fonctionnalités d'Agile Business Suite qui ont joué un rôle clé dans votre décision de migrer vers Agile Business Suite ?*

RT : La fonctionnalité d'intégration particulièrement puissante d'Agile Business Suite a constitué le facteur le plus déterminant. Ces fonctionnalités, qui aujourd'hui sont assurément précieuses, seront d'autant plus importantes pour notre société qui devra faire face aux futurs défis auxquels nous serons confrontés.

DA : *Vous avez mis en production Agile Business Suite 1.2 depuis juin 2009. Qu'en est-il de votre expérience d'utilisation d'Agile Business Suite dans un environnement de production ? Que pensent vos développeurs d'Agile Business Suite Developer ?*

RT : En migrant vers Agile Business Suite, nous avons pu préserver la qualité de notre exploitation, sans compromettre pour autant nos performances. Tous les membres des différentes équipes Unisys avec lesquels nous avons collaboré nous ont fourni une assistance précieuse tout au long du projet. Nous avons particulièrement apprécié la réactivité des ingénieurs Agile Business Suite qui nous ont rapidement fourni les correctifs dont nous avons besoin pour éviter tout retard majeur.

En termes de développement, certains de nos développeurs sont déjà opérationnels et maîtrisent Agile Business Suite Developer. Les autres devraient pouvoir tirer pleinement profit d'Agile Business Suite Developer très prochainement.

DA : *Qu'avez-vous appris lors de ce projet de migration que les autres services informatiques pourraient trouver utile ? Quelles ont-été les activités les plus importantes menées au cours du projet de migration ?*

RT : Nous avons traité la migration de la même façon que tous les autres projets et je pense qu'il s'agit là d'un état d'esprit qu'il est important d'adopter. C'est ainsi que nous avons pu mobiliser les ressources internes, réaliser l'étude de cas et engager le personnel de support d'Unisys nécessaire pour conduire le processus.

Dans l'ensemble, l'activité la plus importante que nous ayons menée fut de permettre aux utilisateurs finaux de tester les applications migrées. Nous avons également utilisé les services et le jeu d'outils Business Application Test Manager (BATMan) d'Unisys pour rejouer l'équivalent de plusieurs jours de transactions EAE dans l'application Agile Business Suite migrée, ce qui nous a procuré une certaine tranquillité d'esprit étant donné que nous étions assuré d'avoir testé la majorité de l'application avant son lancement. De toute évidence, les tests prennent du temps, mais il s'agit d'une étape de la plus haute importance pour garantir une migration en douceur.

DA : *Quels nouveaux développements prévoyez-vous d'effectuer avec Agile Business Suite ?*

RT : Notre principale application de guichet, MOSYS, étant constamment actualisée et étendue pour répondre aux nouvelles exigences des utilisateurs finaux, nous avons prévu dans un premier temps d'utiliser Agile Business Suite Developer pour créer de nouvelles fonctionnalités novatrices qui viendront enrichir MOSYS.

BATMan garantit une transition en douceur vers Agile Business Suite

Proposé en tant que service, BATMan automatise le développement de scripts de test qui capturent les transactions EAE ou Agile Business Suite en temps réel pour vous permettre de tester des applications en ligne avant leur lancement. BATMan relit ces transactions dans un environnement de test cible et offre la possibilité de confirmer toute lecture réussie en confirmant que les résultats correspondent bien aux transactions d'origine. BATMan apporte ainsi une garantie supplémentaire lors de la migration de vos applications de l'environnement de test vers l'environnement de production.

Portez les applications migrées vers de nouvelles sphères grâce à la modélisation dans une optique de réutilisation

Par Grant McCauley, directeur technique en charge du modèle Agile Business Suite, GTC Australie, Unisys Systems & Technology

Les avantages conférés par les fonctionnalités de modélisation UML, notamment celles proposées par Agile Business Suite, sont relativement évidents lorsque vous développez une nouvelle application. Mais il est possible d'en tirer profit avec les applications existantes, plus particulièrement pour ceux qui migrent d'EAE vers Agile Business Suite. Cet article passe en revue des moyens simples et directs pour utiliser les fonctionnalités de modélisation d'Agile Business Suite pour faciliter la prise en charge de vos applications EAE migrées et la réutilisation de votre code EAE migré. En exploitant ces astuces, vous doperez non seulement votre productivité sur le long terme, mais vous bénéficierez également d'une expérience pratique en matière de modélisation tout en travaillant avec les (anciens) systèmes EAE que vous maîtrisez.

Les points importants d'abord ! Lorsque vous migrez vers Agile Business Suite, votre modèle EAE est converti en modèle Agile Business Suite et la plupart des éléments de base sont créés pour vous. Objets, classes, méthodes, attributs... — tous les composants orientés objet indispensables — seront présents une fois la migration effectuée. Étant donné que nous avons abordé les bases des concepts orientés objet dans de précédents articles du Coin des ingénieurs, je ne rentrerai pas dans les détails. Je vous encourage cependant à relire ces articles lorsque vous aurez le temps.

Concernant le modèle Agile Business Suite de votre application, il existe de nombreux moyens de faciliter sa compréhension, sa maintenance, son utilisation et sa réutilisation. Je répartirais ces moyens dans les catégories suivantes :

1. Analyser
2. Organiser
3. Documenter
4. Développer

Examinons chacune de ces catégories plus en détail.

Analyser

Après avoir migré votre modèle vers Agile Business Suite, il peut être judicieux dans un premier temps de revenir en arrière et d'analyser ce dont vous disposez à présent. Il est inutile d'effectuer des modifications par pur plaisir. Prenez simplement le temps d'analyser votre modèle « sous tous les angles ». Cliquez sur les onglets (propriétés, dépendance, etc.) pour explorer les différentes vues disponibles du modèle via Agile Business Suite Developer. Vérifiez les dépendances qui existent, ainsi que la façon dont les propriétés telles que le type, le propriétaire, les héritages et la multiplicité ont été définies pour construire les objets. Vous pouvez également tester la boîte de dialogue Search (Alt+S) et les fonctionnalités de Quick Navigator (Alt+G) pour analyser votre système et votre modèle migrés.

Organiser

Après vous être familiarisé avec l'apparence de votre modèle applicatif dans Agile Business Suite, il est temps de l'organiser. J'entends par organiser la prise de mesures qui faciliteront la compréhension du modèle et éventuellement sa réutilisation et sa restructuration. Une fois encore, vous ne partez pas de zéro, vos zones fonctionnelles et vos activités EAE ayant été transférées sous Agile Business Suite lors de la migration.

Je vous recommande d'utiliser trois outils pour organiser votre modèle : les dossiers, les dictionnaires et les diagrammes. >>

Remise à niveau sur les concepts orientés objet

Pour plus d'informations sur les concepts orientés objet, consultez les documents suivants avant de lire les articles du Coin des ingénieurs :

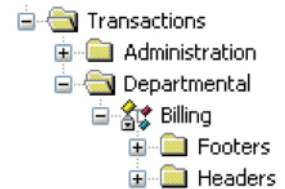
- [Lorsque l'application d'un stéréotype est une bonne chose](#)
- [Encapsulation](#)
- [Polymorphisme](#)



Dossiers

C'est aux dossiers que vous devez vous intéresser en premier lorsque vous organisez votre modèle car ils constituent un excellent moyen de mettre de l'ordre dans votre modèle. Utilisez les dossiers d'Agile Business Suite de la même façon qu'un dossier physique dans lequel vous regrouperez des éléments en rapport les uns avec les autres. À titre d'exemple, vous pouvez souhaiter :

- placer toutes les transactions en rapport avec un secteur d'activité particulier dans un dossier ;
- créer un dossier dans lequel placer toutes vos classes externes, tous les reports ou toutes les transactions administratives ;
- utiliser des dossiers dans un report pour organiser les entêtes et pieds de pages.



Les dossiers sont tellement flexibles que la même entité peut être utilisée dans plusieurs dossiers et que des dossiers peuvent être organisés dans d'autres dossiers.

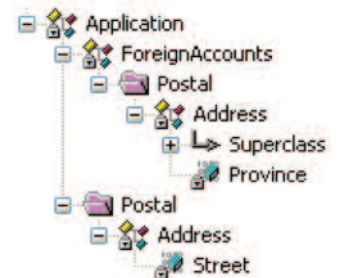


Dictionnaires

Dans Agile Business Suite, le dictionnaire est un type de dossier spécial spécifiquement utilisé pour organiser des objets servant de définitions aux autres objets, notamment les classes que vous voulez réutiliser facilement dans de nombreuses zones de l'application. Contrairement au système EAE, Agile Business Suite propose autant de dictionnaires que nécessaires et ces derniers peuvent être présents aux différents niveaux du modèle (auquel cas ils s'appliquent aux objets quels qu'ils soient définis sous ce niveau).

Pourquoi disposer de plus d'un dictionnaire ? Pour les mêmes raisons que pour les dossiers, excepté que les dictionnaires présentent un avantage supplémentaire : ils facilitent la réutilisation en configurant automatiquement la relation d'« héritage » lorsque le nom d'un attribut ou d'une variable correspond au nom d'un objet figurant dans un dictionnaire.

À titre d'exemple, il peut être judicieux de disposer de plusieurs dictionnaires si vous combinez les fonctionnalités de deux ou trois applications ou si vous souhaitez pouvoir redéfinir certains termes ou définitions dans une zone locale du modèle. Vous pouvez placer un dictionnaire définissant une adresse avec nom de rue, numéro d'appartement, ville, état et code postal dans un emplacement et redéfinir l'adresse pour qu'elle inclue un code postal long, un pays et une province dans un autre emplacement du modèle en rapport avec d'autres pays. Dans chaque emplacement, il serait légitime de se référer à une adresse comme un objet unique et la lier automatiquement à l'objet approprié.

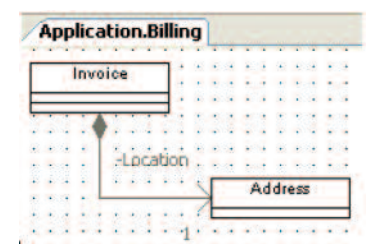
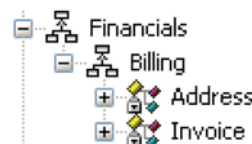


Dans EAE, les définitions figurant dans un dictionnaire sont limitées aux types primitifs tels que les chaînes et les chiffres. Dans Agile Business Suite, il peut également s'agir de classes comprenant des attributs, une logique et même une présentation. Lorsque vous effectuez des entrées dans un dictionnaire, vous fournissez un schéma pour les éléments couramment utilisés dans votre modèle, ce qui peut constituer une première étape vers une réelle réutilisation.



Diagrammes

Les diagrammes constituent un autre outil d'organisation. Comme pour les dossiers, vous pouvez faire glisser des éléments connexes sur un diagramme et créer ainsi une image UML de ce qu'ils contiennent. Même en l'absence de lignes (dépendances/relations) reliant les éléments sur le diagramme, vous avez malgré tout une longueur d'avance car vous disposez d'un regroupement logique d'éléments connexes. Comme le dit le proverbe, « une image vaut mille mots ». >>



Enfin, l'utilisation de l'une de ces astuces d'organisation n'aura aucun impact négatif sur votre application. Il s'agit tout simplement d'outils permettant aux développeurs de mettre un peu d'ordre dans leur monde.

Documenter

Il s'agit tout simplement des actions mises en œuvre pour permettre aux personnes entrant en contact avec votre système de comprendre clairement la fonction de tous les objets, de la logique, etc. Les diagrammes UML ne sont qu'un moyen de décrire un système en créant des vues dans une application et en indiquant les relations structurelles entre les classes.

Agile Business Suite dispose également de fonctionnalités exceptionnelles de « documentation à la volée » qui facilitent l'insertion de commentaires et de descriptions lors de la conception et du développement. Ces mêmes fonctionnalités peuvent être utilisées avec le système migré — considérez-les comme des fonctionnalités de « documentation au fur et à mesure de votre découverte » — pour formaliser vos connaissances institutionnelles sur les différentes fonctions d'un système que vous connaissez déjà parfaitement. Vous pouvez également ajouter une documentation ciblant différents publics. Des informations telles que la fonction métier prise en charge par un objet et les futurs plans prévus pour étendre une fonction peuvent être utiles aux concepteurs. Les utilisateurs finaux tirent également profit de toute documentation leur expliquant comment utiliser un Ispec ou la source des données répertoriées.

Commencez par spécifier une documentation au niveau du dossier et approfondissez en fournissant des détails supplémentaires si nécessaire. Si vous travaillez dans une zone du modèle et que vous découvrez quelque chose d'intéressant, vous pouvez également ouvrir la fenêtre de documentation et noter vos découvertes. Une fonction de type WordPad simplifie le processus au maximum.

Enfin, Agile Business Suite intègre une nouvelle fonctionnalité permettant d'extraire la documentation d'applications. Les pages de commentaire, qui ne sont autres que des guides de référence navigables HTML, fournissent une solution plus puissante et plus conviviale que les sorties standard EAE Print Specification. La gestion de la documentation en ligne s'effectue plus facilement quand elle est automatiquement générée depuis votre système lorsque nécessaire.

Développer

Jusqu'ici, aucune des actions que j'ai recommandées n'a modifié votre modèle applicatif. L'analyse, l'organisation et la documentation se contentent de faciliter sa compréhension. Le développement d'un système existant porte sur l'affinage et l'extension de l'application, ce qui nous mène au cœur de la modélisation dans une optique de réutilisation. Voici quatre étapes que vous pouvez appliquer pour ce faire : supprimer, encapsuler, décomposer et généraliser.

Supprimer

Il s'agit de l'affinage des rôles des entités d'une application en supprimant les fonctionnalités inutiles ou redondantes qui n'ont plus aucune fonction. EAE ne proposant que des entités de haut niveau, vous disposez inévitablement d'opportunités d'affinage. À titre d'exemple, des groupes peuvent avoir été utilisés pour définir des structures sans qu'il ait été nécessaire d'utiliser la fonctionnalité de sérialisation de chaînes ou des chaînes insérables peuvent avoir été utilisées pour des formes sans qu'il ait été nécessaire d'insérer un macroprogramme spécial. La suppression de ces fonctionnalités inutilisées simplifie non seulement la définition d'une entité, mais renforce également sa capacité de réutilisation.

Gardez un œil sur les objets d'Agile Business Suite qui n'appliquent pas leur comportement stéréotypé et transférez-les vers une classe (non stéréotypée) qui leur permettra de tirer pleinement profit de l'orientation objet.

Encapsuler

L'encapsulation facilite la réutilisation en regroupant les fonctionnalités communes et en contrôlant leur accès, ce qui élimine les effets secondaires imprévus. Lorsqu'un objet est correctement encapsulé, il prend la forme d'une « boîte noire » fournissant une interface pour l'utiliser, mais les détails de sa mise en œuvre sont conservés masqués dans l'objet. Avec cette approche, vous pouvez réutiliser l'objet n'importe où dans votre modèle et savoir avec exactitude comment il se comportera. De plus, les modifications internes qui pourraient être apportées à l'objet n'auront aucun impact externe. >>

Commencez par encapsuler les objets en modifiant la visibilité de leurs membres. Par défaut, la plupart des éléments de données sont migrés depuis EAE comme des attributs « publics » et sont accessibles dans n'importe quelle autre zone de l'application. Si vous voulez protéger les données d'une classe Ispec pour qu'elles ne puissent pas être modifiées en dehors de cette classe, définissez leur propriété de visibilité sur « privée ». En modifiant la visibilité des données, toute logique en dehors de cette classe tentant d'accéder aux données désormais internes sera rapidement identifiée car vous obtiendrez une erreur dès que vous tenterez de valider le modèle. L'encapsulation se poursuit en transférant cette logique vers une méthode dépendant du même propriétaire que les données.

Commencez avec un ou deux attributs pour en déterminer l'impact. À mesure que vous parcourez votre modèle, il est possible que des schémas émergent, ce qui se traduit par une réduction du nombre de méthodes, l'élimination de toute duplication du code et le développement de classes plus puissantes.

Voici un exemple : si votre modèle dispose d'une classe Ispec nommée ACCT et que vous définissez l'attribut BALANCE comme attribut privé, il est possible que vous obteniez de nombreux emplacements hors de la classe ACCT présentant une logique comme suit lorsque vous validerez le modèle :

Subtract TRANTOTAL ACCT.BALANCE : Soustraire le total du solde du compte

Il est également possible que vous disposiez d'une logique permettant de vérifier le solde et de signaler une erreur si le solde passe en-deçà d'une certaine limite. Les conditions déterminant si le solde du compte peut être modifié doivent être contrôlées par son « propriétaire », la classe ACCT. En plaçant cette logique dans la classe et assurant sa disponibilité par une méthode, vous ne vous contentez pas de prévenir toute actualisation incorrecte du solde. Vous pouvez simplifier le code dans d'autres zones du modèle et faciliter la modification de la logique d'actualisation en cas de modification ultérieure des contraintes du métier.

Décomposer

Non, cette section ne traite pas de décomposition proprement dite. En informatique, on entend par décomposition le processus permettant de créer des objets plus petits et plus facilement gérables à partir d'objets plus volumineux. Dans EAE, copier une logique d'un Ispec dans un autre — et d'une application dans une autre — était un jeu d'enfant. Avec Agile Business Suite et les concepts orientés objet, l'approche diffère. Les fonctionnalités communes doivent être définies comme des classes et des méthodes qui peuvent être partagées entre une ou plusieurs applications. La complexité s'en trouve réduite et la réutilisation favorisée. La décomposition combine la logique répétitive dans un petit nombre de méthodes et regroupe ces méthodes dans des classes depuis lesquelles elles peuvent être réutilisées.

Prenons pour exemple la gestion des stocks. Lorsque votre système traite une vente au comptant, une vente à crédit ou un retour, le stock de l'article correspondant est actualisé en conséquence (plus ou moins). Il semble donc judicieux de définir une nouvelle classe intitulée StockLevel, assortie d'une méthode nommée UpdateStock permettant de procéder aux opérations d'ajout et de suppression chaque fois que les niveaux de stock changent. De plus, avec cette approche, en cas de modification fondamentale de la façon dont les niveaux de stock sont gérés, la logique n'est modifiée que dans un seul emplacement, mais l'impact de la fonctionnalité est appliqué à l'ensemble de l'application.



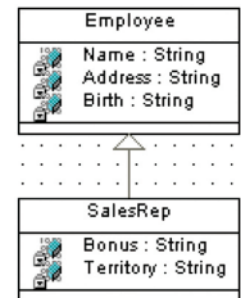
Vous pouvez obtenir des indications concernant les schémas couramment utilisés si vous prenez le temps d'organiser votre système dans un diagramme UML. Les relations vous indiqueront les éléments de données communs, la logique et bien plus encore, soit autant d'opportunités de décomposition. >>

Généraliser

Dernière technique utilisable, la généralisation a pour fonction de simplifier le modèle existant en utilisant l'héritage. Recherchez les caractéristiques et les comportements communs qui sont partagés par plus d'une classe et définissez-les comme une nouvelle classe supérieure. Vous pouvez alors permettre aux objets existants de devenir des classes supérieures héritant de la classe de leurs parents et étendant cette dernière. De plus, en cas de modifications des exigences, il vous suffit de modifier la classe supérieure, les modifications étant appliquées aux sous-classes grâce à la magie de l'héritage.

L'employé constitue l'exemple classique de généralisation. Les employés sont assortis d'un ensemble défini de caractéristiques et de comportements : nom, adresse, date de naissance, etc. Vous pouvez définir certains types d'employés tels que les représentants commerciaux. Ces derniers héritent de toutes les caractéristiques des employés, mais disposent également d'attributs spéciaux qui leur sont propres, notamment la structure des primes qu'ils perçoivent ou leur territoire.

En recherchant des opportunités de généralisation, vous pouvez regrouper des éléments de données communs, une logique et même une présentation dans un groupe généralisé puis définir des spécialisations pour créer des sous-classes.



Portez votre modèle EAE vers de nouvelles sphères

En migrant vers un environnement de développement orienté objet, votre entreprise gagne en productivité.

Procédez comme détaillé dans cet article pour vous familiariser avec Agile Business Suite — tout en migrant vos modèles existants le long du continuum UML.

Silverlight introduit une nouvelle possibilité pour les interfaces utilisateur

En juillet 2008, nous avons consacré une [grande part de Developing Agility](#) aux interfaces utilisateur en publiant plusieurs articles traitant des nombreuses options dont disposent les développeurs pour faciliter l'apprentissage et l'utilisation de leurs applications EAE ou Agile Business Suite. À cette époque, vous disposiez, parmi les options, des générateurs d'interfaces utilisateur d'EAE et d'Agile Business Suite, d'outils client tels que ASP.NET Generator et de commandes personnalisées parmi lesquelles Web Form Renderer. Nous avons également souligné qu'il était possible de créer un générateur personnalisé au moyen de Client Tools Customization Kit ou d'utiliser un générateur conçu par une tierce partie compétente.

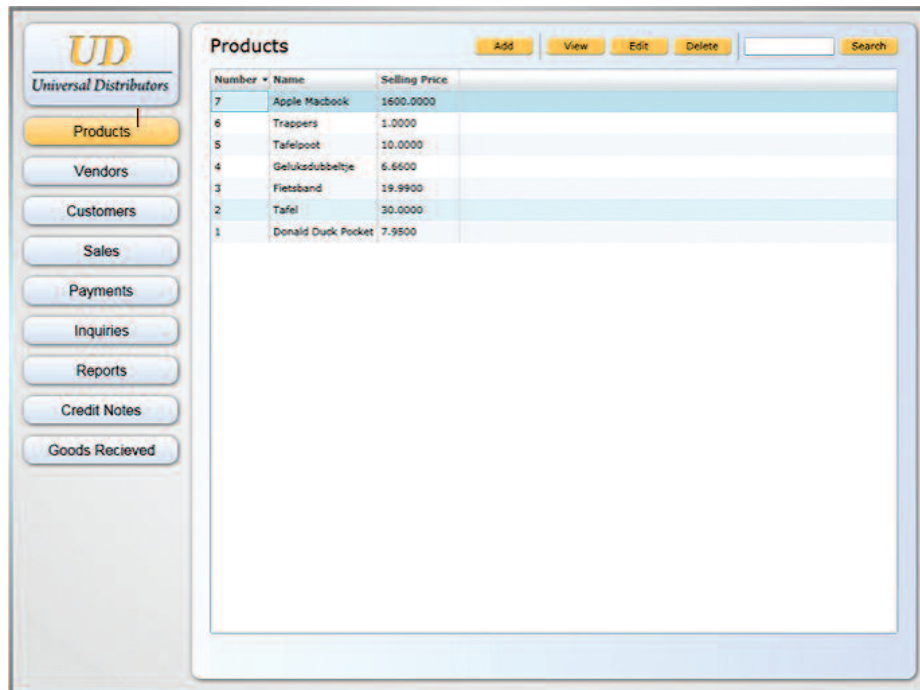
Mais un tout nouvel outil vient de faire son apparition — le plug-in de développement pour navigateur Internet Microsoft® Silverlight™ —, ce qui signifie que vous disposez d'un autre outil pour créer de meilleures interfaces utilisateur graphiques.

Pour plus d'informations sur [Silverlight](#), vous pouvez vous connecter au site Web de Microsoft et même télécharger une copie du kit de développement. Silverlight offre des fonctionnalités aussi nombreuses qu'exceptionnelles qui enrichissent l'expérience de l'utilisateur final. Il est important pour les développeurs EAE et Agile Business Suite de savoir qu'il n'est pas plus compliqué d'améliorer les Ispec via Silverlight que pour d'autres options d'interface utilisateur graphique. Il suffit de définir les transactions que vous voulez améliorer en tant que services Web et de vous mettre au travail. Silverlight est un plug-in Visual Studio. Par conséquent, si vous utilisez Agile Business Suite pour vos projets de développement, sachez que les deux environnements sont étroitement compatibles.

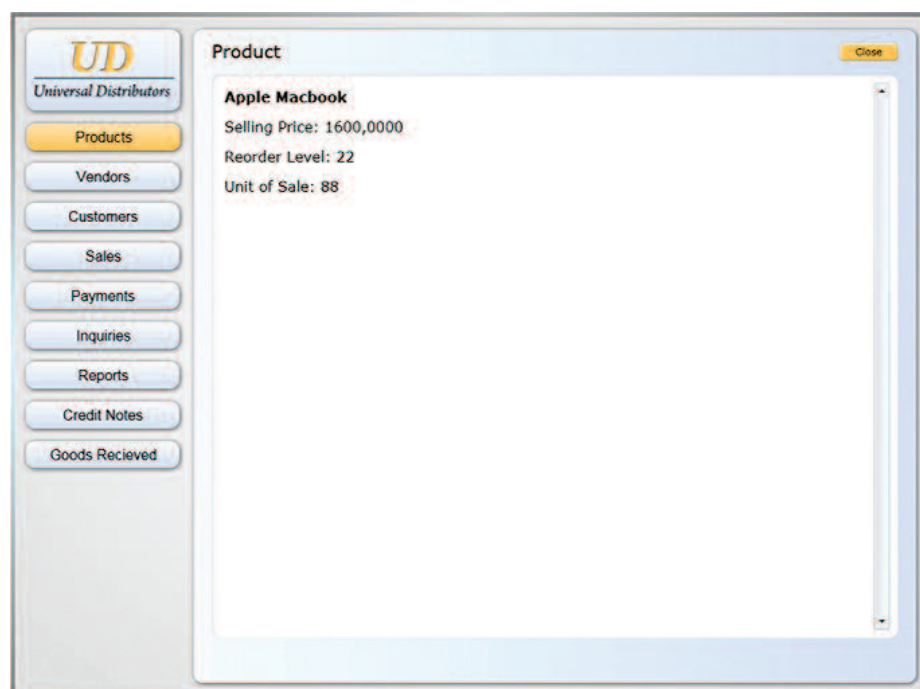
Il est intéressant de noter qu'un groupe d'étudiants néerlandais a récemment mené un projet dans le cadre duquel Silverlight a été utilisé avec l'application « Sample », une amie de longue date. À l'aide d'Agile Business Suite et Silverlight 2.0, les étudiants ont pu transformer une présentation basique en un tout nouveau frontal graphique époustoufflant en l'espace de quelques semaines et sans disposer d'une connaissance approfondie des outils (mais uniquement de connaissances de base sur Visual Studio).

L'équipe a démarré du bon pied en commençant par évaluer l'ancienne interface et en notant toute idée qui permettrait d'actualiser la présentation. Il faut souligner la façon dont ils ont utilisé Silverlight pour combiner les écrans Product List, Details et Maintenance au sein d'un même formulaire. >>

Dans les captures d'écran suivantes, l'utilisateur clique sur le bouton Product pour obtenir la liste de tous les produits sous la forme d'une table impeccable.



Il lui suffit alors de cliquer sur un produit donné pour afficher les détails correspondants, comme représenté dans la capture d'écran suivante. >>



Enfin, les utilisateurs autorisés peuvent assurer la maintenance des informations sur le produit en utilisant le même formulaire.

The screenshot shows a web application interface for 'Product Maintenance'. On the left is a vertical sidebar with the 'UD Universal Distributors' logo and a list of navigation buttons: Products, Vendors, Customers, Sales, Payments, Inquiries, Reports, Credit Notes, and Goods Recieved. The main content area is titled 'Product Maintenance' and contains two sections. The first section, 'Product Information', has a yellow header and contains four input fields: Name, Selling Price, Reorder Level, and Unit of Sale. The second section, 'For Price Changes Only!', also has a yellow header and contains two input fields: Your Name and Reason. At the bottom of the form are three buttons: Submit, Cancel, and Reset.

Dans chacune de ces étapes, seule la partie de l'écran directement concernée est actualisée, plutôt que l'ensemble de l'écran. De plus, les étudiants ont utilisé la même métaphore pour la maintenance fournisseur et client, ce qui permet aux utilisateurs de bénéficier d'une expérience particulièrement cohérente.

À la fin du projet, les étudiants en ont conclu que Silverlight était rapide et facile d'emploi et qu'il constituait un bon moyen d'actualiser l'apparence d'applications plus anciennes.

Quels que soient les outils que votre entreprise choisit d'exploiter — Silverlight ou l'une des nombreuses autres options —, nous vous encourageons à améliorer la fonctionnalité de vos applications à forte valeur EAE et Agile Business Suite avec de nouvelles interfaces utilisateur. C'est un investissement qui, sur le long terme, portera ses fruits.

Calendrier

De nombreux événements passionnants vous attendent. Pensez à consulter régulièrement la section « Webcasts & Events » du site [e-Communauté](#) pour obtenir les informations les plus récentes à ce sujet.

Quoi ?	Où ?	Quand ?
OS 2200 Best Practices Workshops	Bruxelles, Belgique	24 septembre 2009
MCP Best Practices Workshops	Houten, Pays-Bas Bruxelles, Belgique Milton Keynes, R-U Paris, France	29 septembre 2009, 30 septembre 2009, 7 octobre 2009, 13 octobre 2009
Réunion des utilisateurs d'EAE/Agile Business Suite	Paris, France	13 octobre 2009
ClearPath Briefing et OS 2200 Best Practices Workshop	Stockholm, Suède	15 octobre 2009
DACH User Event et OS 2200 Best Practices Workshop	Bâle, Suisse	22 et 23 octobre 2009
Conférence annuelle sur les technologies de UNITE	Hyatt Regency Minneapolis, Minneapolis, MN	8-11 novembre 2009
Symposium consacré à Business Information Server (BIS)	Unisys Roseville, MN	12-13 novembre 2009

Les spécifications sont susceptibles d'être modifiées sans préavis.

© 2009 Unisys Corporation.
Tous droits réservés.

Unisys et le logo Unisys sont des marques déposées d'Unisys Corporation. Microsoft et Visual Basic sont des marques déposées et Silverlight une marque commerciale de Microsoft Corporation. Tous les autres noms de produits et marques mentionnés dans ce document sont considérés comme des marques commerciales ou déposées de leurs détenteurs respectifs.