



Developing Agility

Une lettre d'information destinée aux clients Unisys EAE et Agile Business Suite

Table des matières

- Page 2 **Entretien avec Europaeiske**
Flemming Voigt, responsable informatique de cet assureur danois, nous parle de la migration de son entreprise vers Agile Business Suite.
- Page 5 **UFSS : étude de cas sur la modernisation d'une application EAE**
Une interface utilisateur basée sur des processus, riche en nouvelles fonctionnalités, qui tire parti de services applicatifs éprouvés.
- Page 10 **D'importants changements à venir dans Client Tools**
Unisys opte pour un ensemble Client Tools unique et commun pour EAE et Agile Business Suite.
- Page 12 **Le coin des ingénieurs : prochaines étapes concernant Client Tools**
Conseils pratiques sur l'évaluation de l'impact et des mesures nécessaires pour se préparer à l'ensemble Client Tools commun.
- Page 14 **Calendrier**
Consultez notre calendrier pour connaître les dernières informations sur les événements à venir.



Entretien avec Europæiske

Nous nous sommes récemment entretenu avec Flemming Voigt, responsable informatique chez Europæiske, à propos de la migration de la société vers Agile Business Suite. Basée au Danemark, Europæiske est une société d'assurance spécialisée qui offre une couverture aux voyageurs danois. Lisez cet article pour en savoir plus sur l'environnement informatique d'Europæiske et connaître l'opinion de M. Voigt sur les principaux avantages qu'il espère tirer d'Agile Business Suite.

Developing Agility (DA) : Parlez-nous de l'environnement informatique d'Europæiske.

Flemming Voigt (FV) : Nous disposons d'une application EAE (Enterprise Application Environment) qui prend en charge nos activités d'assurance voyages. Cette application EAE est subdivisée en plusieurs systèmes logiques distincts. Un système gère la gestion des polices d'assurances, depuis l'émission de nouvelles polices, leur éventuelle extension et jusqu'à leur clôture. Nous offrons également un moteur de souscription d'assurance en tant que service Web qui a été intégré aux solutions de plusieurs éditeurs de logiciels, y compris pour les voyageurs au Danemark et en Suède. L'application assure également le traitement et le calcul des commissions des agences de voyage qui vendent nos produits aux utilisateurs finaux.

Nous utilisons également l'application pour calculer et imputer les taxes sur les assurances, assurer les paiements aux ayants droit et alimenter en données consolidées notre entrepôt de données, qui s'exécute sur un serveur Microsoft® Windows®.

L'application EAE se compose de 410 Specs et d'environ 600 rapports.

À l'époque où nous avons migré, nous exécutons l'application sur un serveur ClearPath Libra Model 300 avec 30 MIPS. En janvier 2010, nous avons installé un serveur ClearPath Libra Model 450 avec 40 MIPS, qui était le plus petit modèle disponible. Ce n'est pas la capacité du Libra Model 300 qui a motivé notre décision. Nous voulions disposer d'une machine pour assurer la reprise après sinistre qui pourrait absorber l'ensemble de la charge de production, raison pour laquelle le Libra Model 300 a été installé dans un centre de sauvegarde distant comme solution de reprise graduelle pour la machine de production actuelle.

Nous disposons de deux développeurs EAE/Agile Business Suite et de cinq développeurs .NET qui se chargent du développement des interfaces graphiques, y compris des interfaces utilisateur Web. En interne, nous comptons actuellement une centaine d'utilisateurs accédant à l'application de base, mais depuis que nous vendons de l'assurance sur Internet - où toute la logique métier est développée en LDL et toutes les données sont stockées dans nos bases de données DMSII - nous avons en réalité plus de 1 000 utilisateurs connectés au système. Au cours des derniers mois, nous avons traité plus de 200 000 transactions par jour.

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

FV : En termes d'avantages métier, nous espérons continuer à tirer parti de nos investissements dans nos applications EAE et pouvoir les utiliser dans notre futur environnement informatique. Entre-temps, nous avons également eu recours à LINC/EAE/Agile Business Suite pour réutiliser notre logique. D'abord avec les interfaces de terminal, puis avec les clients graphiques, puis en tant que pages Web et enfin en tant que services Web.

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

FV : Les avantages techniques attendus sont de nature plus stratégique. Nous souhaitons attirer de nouveaux talents en migrant vers un environnement bien connu des développeurs .NET.

En outre, nous espérons que certaines des fonctionnalités orientées objet (OO) disponibles dans Agile Business Suite, en particulier les méthodes, vont faciliter la maintenance de nos applications à l'avenir ainsi que la conception de nouvelles fonctionnalités à l'aide des techniques OO. >>

DA : Quelles sont les fonctionnalités d'Agile Business Suite qui ont influencé votre décision de franchir le pas ?

FV : Avant tout, l'intégration d'Agile Business Suite à la plate-forme Microsoft Visual Studio®. Nous avons également été intéressés par l'outil de contrôle de version, les nouvelles fonctionnalités de programmation orientée objet et la migration sécurisée.

DA : Agile Business Suite 1.2 est en production depuis novembre 2009. Quel est votre retour d'expérience après avoir utilisé Agile Business Suite dans un environnement de production ?

FV : Nous n'avons remarqué aucune différence dans la gestion du runtime par le MCP. Du point de vue de l'opérateur, il n'y a donc eu aucun changement. En fait, nous n'avons observé aucun changement dans les performances d'exécution, mais le temps de réponse sur les transactions en ligne s'est quelque peu amélioré, bien que la différence reste minime.

DA : Quel a été le niveau d'assistance fourni par Unisys avant, pendant et après la mise en production ?

FV : Au cours du projet de migration, nous avons bénéficié de l'assistance d'Unisys Pays-Bas, qui était appuyé par l'équipe de support de l'ACUS pour les problèmes découverts lors du processus de migration et de vérification.

DA : Que pensent vos développeurs d'Agile Business Suite Developer ?

FV : Il ne fait aucun doute que travailler avec Visual Studio est très différent d'EAE, mais après deux semaines, les développeurs se sont habitués à Agile Business Suite. Et l'expérience qu'ils ont acquise au cours de l'atelier en utilisant quelques-unes des nouvelles fonctionnalités d'Agile Business Suite a contribué à les familiariser avec le nouvel environnement.

DA : Avez-vous modernisé l'interface utilisateur graphique de votre application EAE dans le cadre de ce projet ?

FV : Comme nous recourons fréquemment à Component Enabler et que nous avons nous-mêmes développé de nombreuses interfaces utilisateur, les interfaces n'ont pas changé pendant la migration. Nos utilisateurs finaux n'ont donc pas eu connaissance de la migration, tout fonctionnant comme avant.

La migration effective de l'environnement d'exécution s'est limitée à la conception d'une version complète du modèle. Nous n'avons donc subi que 10 à 15 minutes d'arrêt, lorsque la réorganisation de la base de données supprimait certaines structures internes non utilisés par l'environnement d'exécution Agile Business Suite.

DA : Que pensent vos développeurs du nouveau jeu d'outils d'Agile Business Suite ?

FV : Nous avons suivi une formation de base de trois jours pour que tous se familiarisent avec Visual Studio, puis la FAO Agile Business Suite.

Il nous a fallu en fait un certain temps pour maîtriser le développement de frames des reports, tout comme pour travailler avec Painter dans Visual Studio, et nous avons effectivement constaté qu'avec EAE certaines opérations étaient plus simples. Il est en effet moins facile d'obtenir une vue d'ensemble d'une frame lorsque de nombreux attributs sont inclus.

Nous nous sommes tout d'abord familiarisés avec l'écriture de la logique de l'application et nous sommes actuellement proches du niveau d'efficacité que nous avons lorsque nous avons migré de l'EAE.

Du point de vue du développeur EAE, les avantages réels devraient prendre la forme d'un partage des connaissances facilité avec nos développeurs. NET.

DA : Vous disposiez d'un plan de projet détaillé comprenant une semaine d'assistance et de formation sur site. Comment avez-vous réussi à tout faire en si peu de temps ?

FV : En fait, le projet a débuté en juin 2007 avec un atelier d'une semaine qui nous a permis de découvrir Agile Business Suite, mais le projet a été arrêté du fait de certains problèmes de production de notre modèle EAE à cette époque. >>

L'atelier qui s'est tenu début septembre 2009, avec le même instructeur Unisys qu'en 2007, avait un goût de déjà-vu du fait de sa parfaite connaissance de notre environnement. La semaine a donc été très efficace et nous a aidé à nous préparer à la migration réelle.

DA : Plusieurs activités de conseil se sont déroulées hors site. Que pensez-vous de ce type d'organisation ? Savez-vous combien de temps a duré la migration et le nombre de jours qu'Unisys a passé sur le projet ?

FV : Étant donné que nous connaissions le personnel d'Unisys depuis l'atelier, nous avons efficacement franchi les étapes une fois prêts, sans avoir à investir de temps ni d'argent en déplacements.

En outre, l'outil de support à distance, grâce auquel nous pouvons suivre toutes les opérations effectuées par un instructeur Unisys sur notre bureau, nous a permis de gagner en confiance tout au long du processus.

Nous avons passé un contrat avec Unisys pour environ 200 heures de conseil, mais Unisys a également investi un certain nombre d'heures à étudier les meilleures méthodes d'intégration de Microsoft Visual SourceSafe® (VSS) à Agile Business Suite. En outre, nous avons investi environ 450 heures de ressources développeurs pour la gestion de la migration et les tests du système migré.

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

FV : Nous n'envisageons pas actuellement de développer de nouveaux systèmes à l'aide d'Agile Business Suite, mais avec un système de cette dimension, nous ne cessons d'y ajouter de nouvelles fonctionnalités. Nous avons maintenant pris l'habitude de mettre en œuvre de nouvelles fonctions en un temps record. En fait, nous ne planifions les projets qu'un trimestre à l'avance. Nous sommes donc en mesure de nous adapter aux nouvelles opportunités commerciales qui se présentent.

Merci beaucoup à Flemming Voigt de nous avoir fait profiter de l'expérience de son entreprise avec Agile Business Suite.

Les services Unisys synonymes de valeur ajoutée

Comme le mentionne Flemming Voigt dans son entretien, les consultants des services Unisys ont collaboré avec Europaeiske sur les phases clés du projet. Developing Agility a demandé à deux de ces consultants ce qui, selon eux, a contribué à la réussite du projet.

« Pour les projets importants, tels que la migration d'Europaeiske vers Agile Business Suite, une bonne planification est essentielle et une attention particulière doit être apportée au calendrier et à l'impact sur l'environnement de production du client. De plus, les plans doivent couvrir les activités d'Unisys et du client. En ce qui concerne ce projet, la communication exceptionnelle et l'entente mutuelle quant aux actions devant être entreprises, quand et par qui, ont joué un rôle crucial. »

Floor Goedemondt, chef de projet, TCIS, Unisys Pays-Bas

« Au centre de performance et de test d'Unisys, nous avons importé la source d'Europaeiske et testé divers scénarios de contrôle et de gestion des versions et sous-versions. J'ai ensuite passé une semaine sur site au cours de laquelle j'assurais une formation le matin et j'aidais à la migration le reste de la journée. La semaine s'est très bien déroulée et nous avons bien progressé, tant sur la formation que sur la migration. Peu après, Europaeiske a procédé à la migration définitive de son système, qui s'est achevée sans aucun problème digne d'être signalé. Ce projet est selon moi une parfaite réussite ! »

Ching Hong Lam, consultant, TCIS, Unisys Pays-Bas

UFSS : étude de cas sur la modernisation d'une application EAE

Unisys Financial Services System (UFSS) fournit une solution bancaire pleinement opérationnelle, minimisant les risques et sécurisée de bout en bout, conçue pour la gestion des prêts hypothécaires et de l'épargne. Utilisé par un certain nombre d'institutions financières au Royaume-Uni, UFSS est réputé pour sa capacité à maximiser la productivité, à réduire les risques et à maîtriser les coûts, tout en garantissant aux clients une expérience cohérente et rapide permettant de répondre aux exigences de traitement très complexes de ce secteur.

UFSS a été réalisé avec EAE et, comme la plupart des applications EAE, il a été affiné et développé au fil des ans. Aujourd'hui, UFSS représente un capital intellectuel de plus de 20 ans d'expérience et constitue une solution flexible et conforme offrant une plate-forme fiable, efficace et rentable de création et de gestion de prêts hypothécaires et d'épargne.

Ces dernières années, l'équipe de développement UFSS Unisys a mis à jour la solution pour permettre d'accéder aux transactions via Internet. Plus récemment, elle a modernisé l'application pour permettre à UFSS d'exploiter un service Web et pour présenter les fonctions d'UFSS comme des services. Cette étude de cas passe en revue les facteurs opérationnels qui ont influencé les changements et les mesures prises récemment par l'équipe UFSS pour poursuivre la modernisation de cette application vitale.

Expression des besoins clients

La version actuelle d'UFSS (également connue sous le nom d'UFSS Classic) est une solution basée écran passif comprenant plus de 2 000 Specs. Les utilisateurs finaux expérimentés n'ont aucun problème pour naviguer dans l'application à l'aide des identifiants Spec à cinq caractères. Mais pour les nouveaux utilisateurs, la courbe d'apprentissage peut être importante et il n'existe que peu d'indications quant aux transactions qu'un utilisateur doit effectuer et l'ordre dans lequel il doit les enchaîner pour accomplir une tâche donnée.

Suite à des entretiens avec la communauté des clients, l'équipe UFSS a identifié la nécessité d'aider les banques et les sociétés de crédit immobilier à :

- Mettre en œuvre des processus métier qui guident l'utilisateur final tout au long des étapes appropriées, dans le bon ordre, pour qu'il puisse exécuter une opération commerciale en fonction des processus bancaires en vigueur.
- Former les nouveaux utilisateurs rapidement et à moindre coût, qu'ils soient novices ou expérimentés mais ne connaissant pas l'application UFSS.
- Veiller à ce que les interactions avec les clients s'effectuent de manière efficace et cohérente afin de respecter les directives de la Financial Services Authority du Royaume-Uni recommandant de « traiter les clients équitablement ».
- Réduire les coûts sur l'ensemble de leurs opérations.

Compte tenu de la valeur considérable de la solution UFSS, du point de vue de l'investissement financier et des opérations commerciales, il était vital de trouver un moyen de faire évoluer l'application existante pour satisfaire ces exigences. >>

Solution : une interface utilisateur basée sur des processus

L'équipe de développement UFSS avait déjà effectué d'importants travaux de modernisation sur l'application, notamment la conception d'un cadre permettant de s'intégrer dans une architecture orientée services (SOA). Les participants à la présentation EAE lors de la conférence sur les technologies UNITE 2009 ont certainement assisté à la présentation de Tony Bradford en la matière ([reportez-vous au site eCommunity pour visionner les diapositives de Tony](#)). Dans le cadre de ce projet, l'équipe a créé une passerelle d'intégration hôte (Host Integration Gateway ou HIG) qui permet aux applications EAE d'appeler des services Web et des services externes pour la soumission de requêtes de données à partir d'un Ispec EAE. Dans le cadre de ce projet, l'équipe a développé le concept d'un service Ispec qui soit sans état (stateless) qui ne s'appuie pas sur GLB.Work pour stocker les données de session.

Pour répondre aux demandes des clients pour une plus grande facilité d'emploi et d'une meilleure application des processus standard, l'équipe UFSS a pensé qu'il fallait fournir une nouvelle couche d'automatisation des processus exploitant la solution existante. En utilisant la logique applicative UFSS et une nouvelle couche de services métier, l'équipe a créé le concept d'interface utilisateur orientée processus qui regroupe et simplifie plusieurs opérations en les regroupant dans un seul ou un nombre réduit d'écrans, et ce dans un ordre établi mais dépendant des données traitées. Cette approche permet la présentation d'un flux de processus métier qui guide l'utilisateur tout au long d'une opération financière et assure également le suivi approprié et cohérent du processus. La couche de services métier peut également communiquer avec d'autres applications en arrière-plan, leur permettant ainsi d'accéder aux services et données intégrés à cette même interface utilisateur.

Les principaux attributs techniques de l'approche interface utilisateur basée sur des processus sont les suivants :

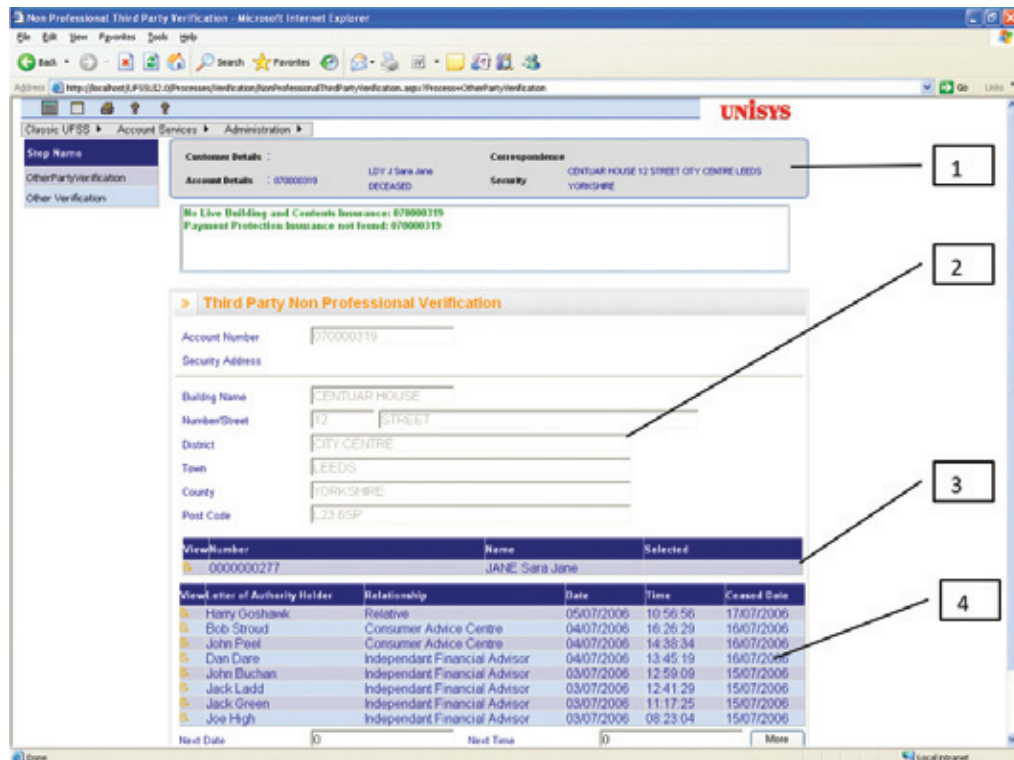
- Les écrans basés sur des processus sont dessinés manuellement à l'aide de la technologie ASP. NET.
- Les commandes AJAX sont appelées lorsque nécessaire pour accélérer l'affichage des données tandis que l'utilisateur navigue sur un écran, ce qui améliore plus encore son expérience globale.
- En arrière-plan, les appels asynchrones sont émis à partir de la couche de l'interface utilisateur pour les services UFSS, via une couche de transport qui s'interface avec Component Enabler. Les résultats de ces appels de services sont utilisés pour alimenter les données à l'écran tandis que l'utilisateur parcourt les étapes d'un processus (par exemple, d'un champ à l'autre).
- Différents champs s'affichent à l'écran selon les données saisies dans les champs précédents.
- L'interface utilisateur basée sur des processus utilisant des plans de site, les clients sont en mesure de mettre en œuvre une sécurité basée sur les rôles .NET pour contrôler l'accès d'autres utilisateurs finaux à certaines fonctionnalités. >>

Mode de fonctionnement de l'interface utilisateur basée sur des processus

Comme c'est souvent le cas, il suffit de quelques exemples pour illustrer le fonctionnement d'un concept technique tel que l'interface utilisateur basée sur des processus.

Exemple 1 : récupération de données

Cet écran rassemble les données requises en appelant divers services UFSS, qui s'exécutent de manière asynchrone et génèrent automatiquement l'écran.



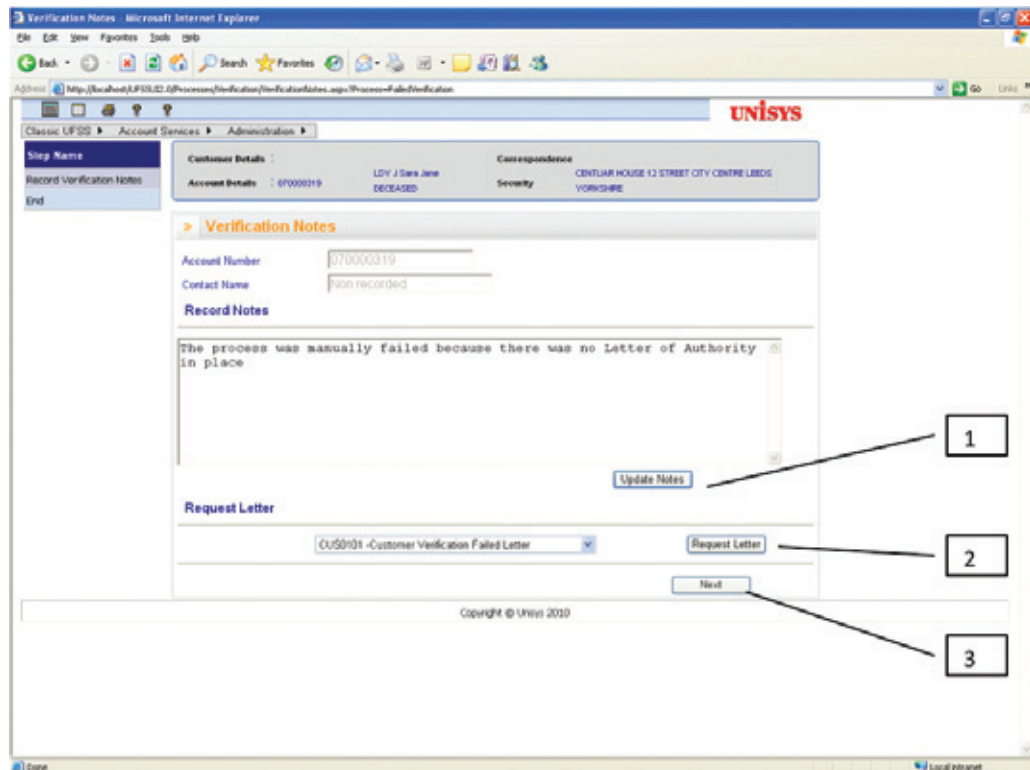
1. Les données contextuelles sont fournies par le service GetAccount (SRV61).
2. L'adresse de sécurité est fournie par le service GetAccountDetails (SRV35), via un fichier de contrôle utilisateur Web .NET qui est placé sur cette page et également réutilisé sur d'autres pages.
3. Les titulaires du compte sont obtenus grâce au service GetAccountHolders (SRV20), une autre commande utilisateur Web .NET réutilisable.
4. La liste Lettre d'autorité est obtenue à partir du service GetLetterOfAuthorityHistory (SRV09). Le bouton situé sur le côté gauche permet de sélectionner une ligne qui, à son tour, appelle un autre service pour obtenir les détails de la lettre d'autorité.

Comme l'illustre l'exemple 1, UFSS présente les données des tableaux sous forme de liste déroulante. Les informations sont transmises par la couche de transport à la couche d'interface utilisateur de sorte qu'elles puissent être liées à la grille de données, sans code C#. Le nombre de lignes est configurable et contrôlé par l'attribut « Nombre de lignes » de la grille, qui est transmis au service Ispec en tant que paramètre.

Une méthode de rendu distincte permet de mettre automatiquement à jour les éléments de données à l'écran à partir des données du modèle Ispec. Il faut pour ce faire s'assurer que les noms des commandes à l'écran correspondent à l'Ispec, ce qui réduit également les efforts de codage. Une méthode similaire est utilisée pour les zones combinées, les cases à cocher et les boutons radio. >>

Exemple 2 : envoi de données

Dans l'exemple 2, les données sont envoyées à UFSS en fonction du bouton sur lequel l'utilisateur clique.



1. Le bouton Update Notes appelle le service SRV11 (une autre commande utilisateur Web .NET réutilisable) pour mettre à jour le bloc-notes de compte. Après avoir cliqué sur le bouton, cette commande est désactivée pour prévenir les mises à jour multiples.
2. La fonction Request Letter appelle le service SRV06 pour demander l'envoi d'une lettre au client. Comme pour le bouton Update Notes, il s'agit d'une commande utilisateur Web .NET et après avoir cliqué sur le bouton, cette commande est désactivée pour empêcher des demandes multiples.
3. Le bouton Next appelle la prochaine étape du processus. Dans cet exemple, il finalise le processus et ramène l'utilisateur à la position de départ.

Bien que la logique UFSS actuelle soit exploitée pour l'interface utilisateur basée sur des processus, l'équipe a dû développer de nouveaux Specs de service, qui sont sans état (stateless) et se comportent comme des boîtes noires. Lorsque les clients UFSS auront migré vers Agile Business Suite, les développeurs UFSS seront en mesure de tirer parti de la possibilité de faire l'invocation d'un Spec par un autre Spec, ce qui évitera le besoin d'avoir deux versions d'une même transaction : une pour l'interface utilisateur classique et une autre pour l'interface utilisateur basée sur des processus. >>

Résultats : une logique métier éprouvée permettant d'activer de nouveaux processus

L'équipe UFSS a récemment présenté les résultats de ce nouveau projet de modernisation d'applications lors de son forum utilisateur et les clients étaient réellement intéressés par les perspectives offertes par l'interface utilisateur basée sur des processus. Ils ont immédiatement noté l'efficacité opérationnelle de cette nouvelle approche qui élimine la nécessité d'impliquer des utilisateurs expérimentés dans les tests de recette utilisateur, ainsi que les avantages qu'elle confère aux utilisateurs novices en termes de convivialité et de réduction des coûts de formation. De plus, l'interface est d'un grand intérêt pour les clients potentiels d'UFSS, qui sont conscients de la valeur d'une interface utilisateur intuitive basée sur le flux réel des processus métier et impressionnés par l'aspect moderne de l'application.

Les processus suivants ont été automatisés et leur déploiement est prévu d'ici peu dans un environnement client réel :

- Recherche de comptes
- Recherche de clients
- Informations récapitulatives sur les comptes
 - Titulaires de compte
 - Avertissement sur les comptes
 - Synthèse des transactions
- Vérification client
- Service de basculement pour permettre l'intégration harmonieuse des transactions classiques avec celles basées sur les processus

Davantage de processus seront encore automatisés dans le futur en fonction des impératifs de temps et de budget et des exigences des clients. L'interface utilisateur basée sur des processus sera disponible dans le courant du 3ème trimestre 2010.

Comme pour l'équipe de développement UFSS, le projet a rapproché les développeurs EAE et .NET du fait des efforts communs qu'ils ont déployés pour présenter une interface utilisateur optimisée tout en utilisant la logique éprouvée de l'application EAE.

Pour toute question concernant les informations présentées dans cette étude de cas, veuillez contacter Tony Bradford, architecte de solutions, Unisys TCIS CET UKMEA, à l'adresse : tony.bradford@gb.unisys.com.

D'importants changements à venir dans Client Tools

Par Diane McGonigle, responsable du programme Agile Business Suite, Unisys TCIS

Unisys a toujours fourni un ensemble Client Tools distinct pour EAE et Agile Business Suite. Chaque ensemble comprend un certain nombre de générateurs de modules clients permettant aux développeurs de créer une grande variété d'interfaces utilisateur final, ainsi que l'infrastructure d'exécution nécessaire à leur déploiement. Bien qu'il existe quelques différences entre le contenu et les capacités d'EAE Client Tools et Agile Business Suite Client Tools, ils présentent également de nombreuses similitudes. En outre, plusieurs versions des mêmes générateurs existent au sein de chaque ensemble Client Tools, que ce soit EAE ou Agile Business Suite.

Malgré les avantages que confèrent des options aussi nombreuses que variées, cela peut être déroutant, même pour les développeurs EAE et Agile Business Suite les plus expérimentés. L'excès n'est pas toujours une bonne chose. Au cours des prochains mois, Unisys mettra en œuvre des changements qui faciliteront l'utilisation des Client Tools par les développeurs et offriront les mêmes fonctionnalités pour les applications EAE et Agile Business Suite. En outre, ces changements nous permettront à l'avenir de vous proposer de nouvelles technologies plus rapidement.

D'ici fin 2010, les actions suivantes seront réalisées :

Évolution vers un ensemble Client Tools unique et commun pour EAE et Agile Business Suite. Ce changement signifie qu'Agile Business Suite Client Tools version 1.2 ou 2.0 pourra être utilisé avec des applications EAE. Chacune des configurations fonctionne aujourd'hui comme depuis un certain nombre d'années. Plus important encore, les futures améliorations Client Tools ne seront intégrées qu'aux versions d'Agile Business Suite et seront compatibles avec les applications EAE.

- Accent sur les versions C# des générateurs clients et suppression progressive du support pour les versions basées sur Java. En juin 2010, une version EAE Interim Correction (IC) mettra en place les changements requis pour Enterprise Application Developer, qui lui permettront d'appeler n'importe quel générateur basé sur C#.
- Fin de support des générateurs reposant sur une technologie non supportée (par exemple, Microsoft Visual Basic® 6 et générateurs de services Web SOAP Toolkit). Des fonctions de remplacement sont actuellement disponibles dans Agile Business Suite Client Tools pour ces deux générateurs.
- Fin de support du générateur ASP. Une technologie plus moderne est disponible avec le générateur de formulaires Web ASP .NET qui inclut la prise en charge d'AJAX et des appareils mobiles.

Common Client Tools (Disponible dès le 31 décembre 2010)

- Générateur de formulaires Web ASP .NET
- Générateur Visual Basic .NET
- Client de présentation
- Générateur de services Web ASP .NET

Une fois le travail terminé, l'EAE Client Customization Kit, qui est vendu sous licence distincte, contiendra les versions finales des générateurs ASP, Visual Basic 6 et de services Web SOAP Toolkit pour tous les développeurs qui souhaitent continuer à utiliser ces outils sans support.

Les deux interfaces d'exécution dans Component Enabler Java et .NET continueront d'être prises en charge. Si vous disposez d'un générateur personnalisé qui utilise l'interface d'exécution Java, vous n'aurez pas besoin de modifier votre mode de déploiement des clients générés. Nous n'assurerons plus le support de la JVM Microsoft (Microsoft n'en assurant plus le support depuis un certain nombre d'années). Nous continuerons à prendre en charge l'environnement d'exécution Sun Java (JRE). >>

Nous continuerons également le support des interfaces d'Enterprise Application Developer et d'AB Suite Developer qui vous permettent d'utiliser des générateurs clients basés sur Java, ainsi que des générateurs plus récents basés sur C#. Par conséquent, si vous disposez d'un générateur client personnalisé écrit en Java, vous pourrez continuer à l'utiliser avec EAE ou Agile Business Suite.

Informations complémentaires

Si vous avez des questions, vous trouverez de plus amples détails à propos de ces changements dans notre livre blanc « Changements apportés aux Client Tools pour Enterprise Application Environment et Agile Business Suite ». Les développeurs qui utilisent EAE ou Agile Business Suite Client Tools doivent télécharger le document à partir du site Web du [Support Unisys](#) ou du site [eCommunity](#).

En outre, l'article [Coin de l'ingénieur](#) de ce numéro étudie différents scénarios pour vous aider à évaluer l'incidence de ces changements sur votre environnement. Dans de nombreux cas, vous ne subirez que peu ou pas d'impact.

Si, après examen de ces deux sources, vous avez encore des questions sur l'incidence de ces changements sur votre organisation, veuillez envoyer un e-mail à : ABSuite@Unisys.com.

Le coin des ingénieurs : prochaines étapes concernant Client Tools

Par John Papachristos, responsable technique Client Tools, GTC Australie, Unisys TCIS
et Nagendra Purushotham, chef d'équipe, GTCI, Unisys Inde

En raison des changements à venir dans Client Tools, les clients EAE et Agile Business Suite concernés doivent prendre des mesures pour veiller à ce que leurs interfaces client continuent à fonctionner. Cet article décrit les différents scénarios et offre des conseils sur les mesures à prendre pour mettre vos applications EAE et Agile Business Suite à niveau.

Scénario 1 : utilisation du générateur ASP

Si vous utilisez le générateur ASP et que vous n'avez pas personnalisé vos clients générés, vous pouvez migrer facilement vers le générateur de formulaires Web ASP.NET. Il suffit d'apporter un changement de configuration à la propriété Component Enabler Bundle pour spécifier le générateur de formulaires Web ASP.NET, puis de régénérer vos définitions de formats ¹ et enfin de déployer les clients basés sur les formulaires Web ASP.NET sur votre serveur Web IIS. Étant donné que ces formulaires ont été conçus en utilisant EAE ou Agile Business Suite Painter, vos utilisateurs finaux les utiliseront exactement comme avant.

Si vous utilisez le générateur ASP et que vous avez **personnalisé** le client généré d'une manière ou d'une autre, la tâche s'avère plus complexe. En effet, vous devrez recréer ces modifications dans le nouvel environnement de formulaires Web ASP.NET. Naturellement, vous devez être familiarisé avec la technologie de remplacement et le langage de programmation, c'est à dire ASP.NET et C#. Une fois ces modifications apportées, les mêmes étapes que celles mentionnées ci-dessus doivent également être appliquées pour appeler le générateur de formulaires Web ASP.NET et déployer les nouveaux clients.

Selon les normes actuelles, le générateur ASP est une technologie ancienne et le générateur de formulaires Web ASP.NET offre un ensemble riche de **fonctionnalités supplémentaires** dont les développeurs EAE et Agile Business Suite aimeront bien profiter, notamment :

- Support d'AJAX
- Support des appareils mobiles
- Utilisation de scripts pour faciliter le déploiement
- Attributs dynamiques
- Support des zones combinées (combo box) automatiques qu'ASP ne génère pas par défaut
- Options de personnalisation utilisant Component Enabler WebFormRender
- Support des outils de test automatisés : Automated Test Tool (avec Agile Business Suite 2.0 uniquement)
- Amélioration des panels et des groupes de boutons (ces fonctionnalités sont disponibles avec Agile Business Suite uniquement car elles utilisent des fonctionnalités seulement prises en charge par le générateur d'écran de la suite Agile Business Suite Developer)

Scénario 2 : utilisation de Visual Basic 6.0

Si vous utilisez le générateur Visual Basic 6², vous devez en toute logique migrer vers Visual Basic .NET. Si les formulaires Visual Basic 6 n'ont pas été personnalisés, le passage à Visual Basic .NET est simple. Mettez à jour la propriété Component Enabler Bundle pour spécifier le générateur Visual Basic .NET, puis régénérez vos définitions de formulaire¹ et enfin déployez les clients Visual Basic .NET sur les postes de travail de vos utilisateurs finaux. Il n'y aura aucun changement du point de vue de l'aspect et de la convivialité pour vos utilisateurs finaux.

Les clients qui ont l'intention de passer à Visual Basic .NET doivent installer Microsoft .NET Framework sur les postes de travail de leurs utilisateurs finaux – ce logiciel doit être installé pour les clients Visual Basic .NET. Les clients qui exécutent Visual Studio 2005 ou une version plus récente seront en mesure de profiter de l'option de publication [ClickOnce](#) pour automatiser l'installation de tous les composants logiciels client nécessaires sur les machines des utilisateurs finaux de leur réseau. >>

¹ À noter que si vous avez effectué des modifications allant au-delà du changement de configuration de la propriété Bundle, vous devrez peut-être régénérer votre application EAE ou Agile Business Suite. Sinon, il suffit de régénérer le bundle.

² Microsoft Visual Basic 6 n'est plus pris en charge par Microsoft depuis plus de deux ans.

Si vous avez **personnalisé** le client Visual Basic 6 généré, l'application de ces modifications dans le nouvel environnement est plus complexe et implique de maîtriser Visual Basic. NET et .NET Framework. Suivez les étapes mentionnées précédemment pour finaliser votre migration vers Visual Basic. NET.

Le générateur Visual Basic. NET offre des **fonctionnalités supplémentaires** que les développeurs EAE et Agile Business Suite peuvent utiliser, notamment :

- Scripts pour faciliter le déploiement
- Attributs dynamiques
- Options de personnalisation utilisant Component Enabler WinFormRender
- Support d'outils de test automatisés (avec Agile Business Suite 2.0 uniquement)
- Amélioration des panels et des groupes de boutons (ces fonctionnalités sont disponibles avec Agile Business Suite uniquement car elles utilisent des fonctionnalités seulement supportées par le générateur d'écran de la suite Agile Business Suite Developer)

Scénario 3 : utilisation du générateur de services Web SOAP Toolkit

Si vous utilisez le générateur de services Web SOAP Toolkit, vous devrez migrer vers le générateur de services Web ASP. NET. Comme il est décrit dans le scénario 1, il suffit de modifier la configuration de la propriété Component Enabler Bundle pour préciser le nouveau générateur, puis de régénérer vos services Web à l'aide du générateur de services Web ASP.NET et enfin faire le déploiement sur le serveur Web IIS.

S'agissant de la création de services Web à partir d'EAE ou d'Agile Business Suite, vous devez tenir compte de l'impact potentiel de ce changement sur les utilisateurs de votre service. Par exemple, les services Web ASP.NET utilisent des structures plus récentes, de type complexes, qui sont rarement créées par le générateur SOAP Toolkit. Il peut donc être nécessaire de modifier l'application qui appelle ou utilise le service Web pour permettre la récupération et la définition des propriétés dans les types complexes. Ces considérations feront l'objet d'une documentation supplémentaire dans un guide pratique qui sera disponible à l'introduction des modifications aux Client Tools.

Les développeurs EAE et Agile Business Suite apprécieront l'amélioration des options de configuration et de déploiement fournies avec le générateur de services Web ASP.NET, ainsi que la possibilité de mieux sécuriser les services Web en utilisant l'infrastructure WS-Security de Microsoft. Et la solution continuera d'inclure l'outil Discovery facilitant le processus de test des services Web.

Si vous utilisez Business Integrator pour accéder à des services Web basés sur SOAP Toolkit, remplacé par les services Web ASP .NET, vous devrez mettre à jour vos spécifications de composants pour référencer les nouveaux services Web ASP .NET. Dans la plupart des cas, vous devrez également modifier la logique du script de spécification des composants pour gérer les types complexes définis dans les services Web ASP .NET.

Scénario 4 : utilisation de la version Java d'un générateur, migration vers la version C

Comme indiqué dans un autre article de ce numéro de *Developing Agility*, les versions Java des générateurs Client Tools finiront par disparaître. Par conséquent, vous devez envisager de migrer vers la version C# au cours des prochains mois. Les utilisateurs Agile Business Suite peuvent procéder au changement immédiatement. Les clients EAE seront en mesure de procéder au changement lors de la mise en œuvre de l'IC 3270, qui introduit le support des générateurs basés sur C# et sera disponible en juin 2010.

Ce passage à un générateur basé sur C# est fait par un simple changement dans les propriétés de configuration du dossier Component Enabler Bundle. Au lieu de spécifier le nom de classe Java pour le générateur basé sur Java, indiquez le nom du générateur basé sur C# (remplacez par exemple `com.unisys.jellygen.GenerateFormASPdotNET` par `GenerateFormASPdotNET.dll`).

Remarque : dans Agile Business Suite System Modeler, une liste déroulante des générateurs standard basés sur C# sera disponible pour faciliter ce changement. >>

Entre autres avantages, la migration vers des générateurs basés sur C# plutôt que vers des générateurs basés sur Java n'exige par l'installation de Java (JRE et SDK) dans votre environnement de développement standard, sauf si vous souhaitez générer un client basé sur Java ou que vous disposez d'un générateur personnalisé basé sur Java.

Scénario 5 : utilisation d'un générateur personnalisé basé sur Java

Si vous disposez d'un générateur personnalisé basé sur Java, vous pouvez continuer à l'utiliser car nous n'allons pas supprimer l'environnement généré par Component Enabler pour Java ou l'interface du runtime Java du Component Enabler. Il convient de noter que nous ne fournissons pas de support pour les générateurs personnalisés, sauf en cas de problème particulier avéré dans notre interface ou environnement client.

Scénario 6 : utilisation de client personnalisés qui n'ont PAS été créés via un générateur Client Tools

Certains utilisateurs ont écrit des modules clients personnalisés pour leurs applications EAE sans utiliser Client Tools. Ces utilisateurs pourront continuer à fonctionner sans changement, les modifications Client Tools à venir n'ayant pas d'impact sur l'interface du runtime de Component Enabler.

Restez à l'écoute pour plus d'informations

Nous préparons une série de guides pratiques détaillant les étapes de la migration vers les nouvelles technologies de génération de clients. En outre, le Guide de l'utilisateur Component Enabler explique comment utiliser tous les générateurs, y compris les générateurs de formulaires Web ASP.NET et Visual Basic .NET.

Si vous avez des questions concernant les informations figurant dans cet article ou sur l'incidence de ces changements sur votre organisation, veuillez envoyer un e-mail à : ABSuite@Unisys.com.

Adieu à la JVM de Microsoft

Microsoft a cessé le support de son produit JVM il y a plusieurs années et cet environnement ne fonctionnera plus avec Client Tools d'ici fin 2010. Les clients utilisant la JVM de Microsoft doivent immédiatement prévoir d'utiliser les interfaces alternatives que nous prenons en charge.

- Si vous utilisez actuellement la JVM de Microsoft pour générer des applications client Component Enabler, vous devez passer à la JRE de Sun dès maintenant. Vous devez également prévoir de passer aux générateurs basés sur C# d'ici la fin de 2010.
- Si vous utilisez la JVM de Microsoft pour le runtime, vous devez prévoir d'utiliser les interfaces JRE de Sun ou .NET Runtime, en fonction de la technologie utilisée par votre application cliente.

Calendrier

Quoi ?

Conférence annuelle sur les technologies [UNITE](#)

Où ?

[Baltimore Marriott Waterfront Hotel](#),
Baltimore, MD

Quand ?

Du 23 au 26 mai 2010

Les spécifications sont susceptibles d'être modifiées sans préavis.

© 2010 Unisys Corporation.
Tous droits réservés.

Unisys, le logo Unisys et ClearPath sont des marques déposées d'Unisys Corporation. Microsoft, Visual SourceSafe, Visual Studio et Windows sont des marques déposées 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.