Square des Utilisateurs

Modèles de conception, Noyaux techniques applicatifs et FrameWorks

Composants des processus métiers

Antoine Clave nous fait part dans cet article de son expérience de consultant objet dans la mise en oeuvre des composants réutilisables qui participent aujourd'hui à l'informatisation des processus métiers. Il nous en présente les différents niveaux, du composant technique au modèle prédéfini véhiculant la connaissance métier.

Introduction

Plusieurs Entreprises ont entrepris la migration de leur Système d'Information, depuis une architecture centralisée vers une architecture décentralisée de type Client / Serveur ; ceci entraîne pour la Direction des Systèmes d'Information l'organisation du passage d'un développement structuré cartésien à une démarche objet qui modifie profondément tout le cycle d'un projet.

Dans ce cas les Études doivent :

- choisir un mode de conception,
- mettre au point ou choisir un nouveau processus,
- choisir un outil d'analyse, de conception et de génération de code,
- sélectionner un (ou des) outil de développement,
- former les équipes,

en bref dégager un nouveau processus complet de mise au point des applications spécifiques (dont les standards de développement ne sont qu'un aspect très partiel),

… et choisir des prestataires externes capables de leur éclairer les obstacles habituellement rencontrés avec la nouvelle démarche et l'outil choisi, et de les guider. Ces consultants externes interviennent souvent très en amont dans le projet, et proposent un accompagnement sur une bonne part de son cycle de vie.

Les Études se voient en outre proposer par des SSII spécialisées des compléments logiciels à l'outil de développement, désignés sous le terme générique de « FrameWorks », constitués de bibliothèques de classes prêtes à l'emploi et censées réduire le travail de réalisation. Les SSII qui proposent ces compléments sont plus ou moins partenaires de l'éditeur de l'outil de développement, à moins que ce ne soit l'éditeur lui-même de l'outil qui les propose en supplément. Le prix de ces compléments logiciels peut s'avérer élevé, que ce soit en coût d'acquisition ou en coût de mise en œuvre.

Gestion des interventions des consultants externes, gestion des différentes acquisitions et investissements logiciels : la tâche de la maîtrise d'œuvre ne s'en trouve pas facilitée. Il lui faut une vision globale, suffisamment détachée pour lui permettre d'arbitrer exclusivement en faveur de ses intérêts les différentes options qui se présentent.

La tâche est d'autant plus difficile que :

- A part l'évaluation régulière des ressources consommées et de celles encore disponibles, les indicateurs fins permettant le suivi et le pilotage des projets ne sont pas encore vraiment popularisés ;
- Dans le cas où la fonction d'assistance à la Maîtrise d'œuvre n'est pas vraiment remplie, les compétences en matière d'objet qui lui permettraient de juger, d'apprécier la validité des différentes démarches ne sont pas encore acquises, et (au moins au début du projet) sont

détenues par des intervenants externes qui précisément sont parties prenantes dans le projet, ne serait-ce qu'à titre de fournisseur des outils ci-dessus mentionnés.

Le but de cet article est de tenter d'examiner les apports de ces « FrameWorks », Noyaux Techniques Applicatifs (NTA) et démarches d'aide à la conception que sont les MODELES, puis de dégager leurs enjeux, d'apporter un petit éclaircissement sur ces bibliothèques et la manière dont elles pourraient s'insérer effectivement dans une démarche fructueuse et surtout auditable, traçable et maîtrisable..

Noyaux techniques applicatifs (NTA)

Définition

On peut regrouper dans cet ensemble des classes qui, liées par héritage, réutilisables, ont pour but de fournir des fonctionnalités générales utilisables par tout progiciel, de tout métier. L'exemple type est la bibliothèque standard C++ avec ses classes flux instream, iostream, … On trouve également des classes dédiées à la gestion de chaînes de caractères, de piles, de tableaux, de vecteurs, de connexions TCP-IP, … Parmi ces services il en est un particulièrement sensible : la gestion du transactionnel avec la base de données. Bref elles fournissent des fonctionnalités transversales d'usage courant que le développeur n'aura pas à développer, et réutilisables dans tout applicatif spécifique développé au sein des Études.

Elles sont le prototype même des classes réutilisables. De plus, quel que soit le développeur, elles permettent de s'assurer que les tâches de bas niveaux faisant appel aux services du système seront réalisées de manière homogène : ce n'est pas un mince avantage et le chef de projet appréciera.

Enjeux fonctionnels

Les NTA devant s'adapter à tout type d'application et à tout type d'environnement, il peut être sage de les spécialiser afin de les adapter au contexte précis de l'entreprise : en relation avec les équipes systèmes et réseaux, ou en collaboration avec un Architecte Logiciel orienté objets, membre du support technique, et en collaboration avec le DBA applicatif, il est intéressant de définir les composants qui devront être exclusivement utilisés par les développeurs.

Il convient donc d'examiner les questions suivantes :

- Quels sont les services offerts par l'outil ? Ces services sont-ils suffisants ? Quels sont ceux qui, n'étant pas nécessaires, devront être prohibés ?
- Quels ont ceux que nous devons acquérir pour compléter la palette disponible ?
- La palette étant constituée, quelle spécialisation faudra-t-il opérer pour en améliorer la souplesse d'utilisation par les développeurs ?
- Répond-elle à des exigences qualité et a-t-elle été éprouvée ? Bénéficie-t-elle d'un solide retour d'expériences ?

Constatons simplement que cette démarche est déjà éloignée d'une simple acquisition et utilisation immédiate de ces fameux « composants sur étagère ».

Enjeux conceptuels

La première origine des NTA est le développement en interne : cette source est, bien entendu, fructueuse mais n'est envisageable qu'au cours de reprises d'expériences, de « débriefings », de capitalisations et donc à la suite d'une pratique de l'outil déjà importante au sein des études. Elle n'entre donc pas dans le cadre de notre hypothèse.

En principe l'outil de développement fournit des classes dédiées à ces tâches élémentaires de bas niveau. Un fournisseur spécialisé se doit d'offrir des facilités supplémentaires et de le démontrer. Un éditeur prend évidemment en compte l'OS et le SGBDR hébergeant les données manipulées par son outil. Mais devant mettre à disposition les richesses du système d'exploitation, son souci est avant tout technique. Il ne peut et ne doit surtout pas se soucier de l'application qui les utilisera :

le fonctionnel n'est pas la préoccupation de l'auteur d'un outil de développement enrichi d'un NTA et il doit en conséquence limiter au maximum les hypothèses sur la mise en œuvre ultérieure. Le cahier des charges est compréhensible par un informaticien orienté système ou Bases de Données et ne présente aucun vocabulaire métier.

Il convient donc de normaliser, spécialiser et, seulement après examen, de compléter par une acquisition à l'extérieur.

Un NTA est donc un moyen d'assise technique sûre, normalisée, éprouvée, mettant à disposition de notre application les ressources du système, et ce de manière suffisamment normalisée pour en faciliter la maintenance. Il n'impose aucune contrainte ni sur le fonctionnel, ni sur l'architecture globale de l'applicatif. Cette assise est suffisamment étroite pour offrir une bonne visibilité sur les échanges de l'applicatif et pour être facilement maintenable. Rappelons enfin que la performance de ces tâches a un impact fort sur la performance globale de l'application.

On peut donc dire que l'acquisition d'un Noyau Technique Applicatif requiert autant d'attention que celle d'un utilitaire ou d'un outil de développement.

FrameWorks

Définition

Ensemble de classes de haut niveau, contribuant à réaliser le fonctionnel de l'Entreprise, un FrameWork est dédié à un secteur d'activité, à une branche économique. Ces classes prennent en charge des tâches spécifiques et récurrentes, bien identifiées comme faisant partie du métier de l'entreprise. Elles sont donc de très haut niveau et s'intercalent dans une description très orientée métier de l'application. Elles doivent mériter le nom de **Composant métier**. Les concepts qu'elles mettent en œuvre sont des concepts métiers, et pas des concepts informatiques. Le cahier des charges de tels composants doit être intégralement appréciable, lisible, compréhensible par un expert du métier de l'entreprise, avec simplement le bagage informatique minimum de l'utilisateur averti.

Nous citerons comme exemple les composants destinés à la modélisation financière, les composants dits décisionnels utilisés lors de la construction d'un Data Warehouse, ou encore les composants graphiques spécialisés destinés aux bureaux d'études.

Un autre exemple pourrait être les « moteurs d'inférence » fournis par certains éditeurs et dont le principe est le suivant : les développeurs traduisent les fonctions à réaliser en une suite de règles et de données de gestion dans un langage plus souple que le langage de développement. On pourrait interpréter cette solution comme une tentative de l'éditeur de prendre pied dans le métier concerné par le projet sans avoir à en connaître vraiment les règles.

Enjeux fonctionnels

Réalisant un fonctionnel fort, ces classes imposent leur vision du problème à résoudre. La moindre des questions à se poser sera celle de l'adhésion de l'Entreprise à cette vision. Et inversement de la compétence de l'éditeur sur le secteur d'activité économique de son client :

- Est-il au fait des problèmes fonctionnels à résoudre ?
- A-t-il déjà collaboré à des projets de même ampleur chez des entreprises de la même branche ?
- Où a-t-il eu à résoudre ce même type de tâches ? (la gestion de stock par exemple chez un VPCiste, chez un grossiste pharmaceutique, … Certaines tâches métiers, bien que spécialisées, sont néanmoins transversales).

Le meilleur moyen de s'en rendre compte sera de vérifier la capacité du commercial à appréhender, à porter jugement, à conseiller sur la description du fonctionnel, à repérer les trous, les imprécisions, les décisions prises trop tardivement, … Ce qui évidemment pose des problèmes de confidentialité qu'il faut prendre en compte.

Les questions à se poser porteront sur le domaine fonctionnel couvert par ces composants, sur la bonne adéquation des hypothèses faites par le concepteur, et donc sur le degré de spécialisation à opérer dans le cadre de l'adoption par l'Entreprise.

Enjeux conceptuels

A la différence d'un NTA, le FrameWork impose une architecture à l'applicatif. Alors que le NTA fournit une assise horizontale, par nature basique, sur laquelle on assoira toute l'application, le FrameWork se présente comme une colonne verticale autour de laquelle on organisera tous les développements ultérieurs. Il s'intercale dans le projet entre les fonctions dites de présentation (les IHM, les éditions …) et les fonctions dites de persistance (gestion des sauvegardes, donc des accès à la base de données …). Il impose donc son organisation, son partitionnement, et la réutilisation ne sera pas une réutilisation de code, mais une réutilisation de conception, ce qui signifie qu'on réutilisera une architecture qui appellera un code spécialisé, alors que dans le cas d'une boîte à outils on écrira une architecture qui appellera du code existant.

Du fait de ces hypothèses très fortes faites par l'auteur du FrameWork quant à l'architecture et au fonctionnel, il convient d'être très vigilant sur la question de la pérennité de ce genre d'outil :

- Quelle est la pertinence des choix fonctionnels effectués ?
- Quelle est l'assise de l'éditeur ?
- Quelle est la taille des équipes affectées par l'Éditeur à la maintenance ?
- Quel est le rythme des mises à jour ?
- Quelle est la disponibilité du support en ligne ?
- Quels sont les modèles de conception sur lesquels ils s'appuient ?

Autant de questions qui seront appréciées aussi bien au sein des Études pour les plus techniques d'entre elles qu'auprès des utilisateurs qui devront être consultés pour celles ayant trait à la partie fonctionnelle, avec d'autant plus de difficultés que ces derniers n'y voient souvent que de la technique informatique (de la plomberie logicielle).

Enfin, l'Architecte Objet devra veiller à minimiser le couplage entre le FrameWork et l'applicatif : une nouvelle version avec modification de l'interface ne devra entraîner que le minimum de mise à jour du code développé en interne. De ce fait il faudra avoir une connaissance approfondie des modèles sur lesquels ils sont basés et des APIs apportées.

On peut donc dire que l'acquisition d'un FrameWork requiert autant de vigilance que l'acquisition d'un progiciel complet.

Modèles de conception

Un modèle de conception n'est pas un produit vendu par un éditeur, mais un guide pour la mise en œuvre de solutions simples à des problèmes de conception récurrents. Un MODÈLE est mis au point à la suite d'une réflexion qui, ayant repéré la répétition de problèmes identiques, répétera et donc systématisera des solutions. En dépit de son nom, cette démarche peut être pratiquée dès la phase d'analyse.

Les modèles de conception peuvent faire partie de la panoplie des prestations purement intellectuelles proposées par les consultants. Ils sont utilisés dans la phase de définition de l'architecture applicative, avant celle de réalisation (codage).

C'est donc un point de vue qui intéresse l'Architecte objet (dans la mesure où cette fonction existe sur la plate forme), opérant dans une démarche de qualité. Il concourt à augmenter la réutilisabilité de l'implémentation, et se traduisant par une diffusion large des mêmes solutions au sein de l'équipe de développement, à une meilleure homogénéité des livrables.

Sans entrer dans une étude détaillée, examinons juste un exemple, celui de l'implémentation d'un modèle des CLIENTS de l'Entreprise : chaque service de l'Entreprise a une vision du même individu (instance) de la classe CLIENT, vision qui n'est pas forcément en rapport avec celle qu'ont les autres services : ainsi il existe une vue logistique du CLIENT (description des tournées de livraison, nombre de livraisons hebdomadaires), une vue COMMERCIALE (Contact, engagement), une vue COMPTABLE (références bancaires, numéro du compte). Chacune de ces vues sera implémentée par une classe (ClientLOG, ClientCOMM, ClientCOMPT) descendant de la même classe ancêtre CLIENT.

Lors de l'instanciation (création d'un individu), il faudra utiliser l'une de ces trois classes (en fonction du profil de l'utilisateur) et initialiser les propriétés adéquates. Que se passera-t-il si on crée un nouveau service administratif dans l'Entreprise, nécessitant la définition d'une nouvelle vue de chaque concept métier ? Faudra-t-il modifier l'architecture, donc le code, et donc procéder à un nouveau déploiement du logiciel complet ?

Une solution plus simple consiste à confier à une classe spécialisée, FABRIQUE, le soin d'appeler le constructeur[1] adéquat ; ainsi, après que le serveur applicatif aura pris connaissance du contexte utilisateur (droits, profil), il demande à FABRIQUE de mettre à disposition une instance de la bonne classe. Il en résulte que l'apparition d'un nouveau service dans l'entreprise, donc la nécessité de définir et d'implémenter une nouvelle vue, ne se traduira que par l'évolution d'une seule classe : la classe FABRIQUE, et la création de la nouvelle classe vue ClientXYZ.

Avantage important : il n'y aura qu'un impact très limité sur le code ; on maîtrisera mieux l'adaptation du logiciel à l'évolution du métier de l'Entreprise.

Cette démarche d'amélioration du « design » de notre application est l'œuvre d'un spécialiste objet ; soit le Chef de Projet s'en charge, le plus souvent au cours de la phase de conception, soit un Architecte dédié en renfort d'équipe se charge d'examiner les livrables des équipes et de proposer des solutions équivalentes plus maintenables.

La proposition d'intervention extérieure se traduira non pas par la livraison d'un produit tangible à utiliser, mais par une prestation intellectuelle d'accompagnement dans un domaine que la Maîtrise d'ouvrage n'a pas encore pleinement apprécié. De plus, confrontée en prise directe avec les travaux des équipes, cette intervention risque de soulever des problèmes humains qu'il faudra gérer ; les questions à se poser seront, outre la compétence forte en architecture objet, l'habitude que le consultant qui nous sera envoyé doit avoir de s'insérer dans une équipe, juger son travail et proposer des améliorations sans déclencher de conflit.

[1] *constructeur : Opération qui crée un objet et initialise son état.*

Récapitulatif

	NOYAU TECHNIQUE APPLICATIF	FRAMEWORKS	MODÈLES DE CONCEPTION
Nature du produit	Ensembles de classes	Ensembles de classes	Conseils en architecture logicielle visant à harmoniser, optimiser et normaliser les solutions d'implémentation
Mis en œuvre par	Instanciation. Fournit une assise horizontale.	Spécialisation (dérivation). Fournit une structure verticale.	Implémentation
Spécificité à l'environnement métier de l'entreprise	Aucune. Le fournisseur connaît parfaitement l'outil de développement.	Très forte. Le fournisseur doit a priori avoir une connaissance étendue du fonctionnel de son Client	Non
Spécificité à l'environnement technique de l'entreprise	Très forte : mettent les ressources systèmes à disposition du progiciel.	Aucune	Aucune.
Validation / évaluation conduite par …	Personnel informaticien technique qui appliquera des critères mécaniques	Personnel utilisateur qui appliquera des critères fonctionnels (règles de gestion, règles métiers …)	Architecte logiciel, chef de projet, responsable qualité.
Sous la responsabilité de :	Chef de projet informatique	Chef de projet utilisateur	Directeur des Études, DSI.
Retours d'expériences utiles :	N'importe quelle entreprise de taille comparable utilisant les mêmes solutions	Entreprise confrère, confrontée au même problème fonctionnel, donc a priori du même secteur d'activité.	Toute Entreprise
Conséquences sur les choix de développement	Au niveau de l'accès aux ressources	Sur toute l'architecture du projet : impose son architecture à l'application.	Conception, réalisation, maintenance
Bénéfices à en attendre	Les ressources sont utilisées de manière homogène : optimisation des performances.	La réalisation de tout le fonctionnel est menée de manière homogène ; optimisation de la couverture des besoins.	Meilleure vision globale du projet, et à termes de l'ensemble des projets. Identification aisée des modules logiciels.
Points à surveiller	Performances, adéquation à l'environnement technique. Assise taille et pérennité	Choix fonctionnels, adéquation au métier. Assise taille et pérennité de l'éditeur. Couplage avec le	Compétences …

	de l'éditeur. Couplage avec le progiciel à développer.	progiciel à développer.	
Points qualité impactés	Performance, sûreté des accès.	Règles fonctionnelles.	Architecture maîtrisée.

Conclusion

Les négociations commerciales avec un éventuel fournisseur doivent conduire à une vision précise des produits proposés. Les propositions ne donnent pas toujours (jamais) du produit ou des prestations à livrer une description apte à éclairer la Maîtrise d'œuvre : il faut insister pour qu'on décrive le produit en terme de services, d'Architecture, avec des schémas concrets, des exemples tirés du fonctionnel réel de l'Entreprise, et permettant de situer le niveau d'insertion de l'outil dans la démarche du projet. Le produit proposé est-il un FrameWork stricto sensu, ou un Noyau Technique ? Ou un mix des deux ?

Il faut que la Maîtrise d'œuvre conserve la pleine maîtrise de ses choix et ne cède surtout pas à la tentation de déléguer à des consultants externes la prise de décision, même technique, d'acquisition d'un NTA ou d'un FrameWork. Les consultants ne sont que des consultants. En particulier, l'ouverture d'un nouveau poste au sein d'une DSI devrait se traduire dans un premier temps par l'arrivée d'un consultant externe qui va l'occuper, le temps d'embaucher un permanent salarié : cette situation doit être temporaire ; la faire durer plus de 3 mois serait une erreur.

Un moyen efficace de roder le département des Études aux changements induits par la migration mentionnée en introduction et de tester la qualité de l'accompagnement offert par les consultants externes serait de définir (un) des projets pilotes, de petite taille, permettant de juger la qualité humaine des consultants, mais aussi la capacité de l'équipe à faire la place à de nouvelles technologies.

Il convient d'apprécier la qualité non pas uniquement sur la compétence en objet, mais sur la capacité à expliquer simplement les problèmes, et la capacité à dégager et proposer **plusieurs** solutions possibles à un problème.

Antoine Clave
Consultant objet