

SPEM ou la modélisation des processus logiciels

Le Software Process Engineering Metamodel permet de décrire le processus de production des logiciels et de faciliter le partage des informations.

Dans l'industrie du logiciel, de nombreux concepts et méthodes de développement existent. Les développements logiciels devenant de plus en plus complexes, les équipes doivent désormais intégrer de nombreuses informations autant sur la technologie utilisée (Java, .NET, ...) que sur l'outil de développement lui-même, mais également sur l'organisation adoptée pour travailler (méthodes agiles et itératives).

Les organisations traditionnelles et agiles sont toutes deux confrontées à la difficulté du partage d'informations :

- peu de documents sont centralisés, il est ainsi fréquent que plusieurs personnes aillent chercher le même document et ne tombent pas sur la même source ;
- il est difficile d'intégrer les différentes méthodes et processus qui sont présentés sous un format à chaque fois différent ;
- les approches utilisées lors d'un projet sont difficilement réutilisables à cause de la nature spécifique de chaque projet.

SPEM Software Process Engineering Metamodel - que l'on peut traduire par méta-modèle d'ingénierie des procédés logiciels - est un métamodèle (ou modèle décrivant les concepts) visant à décrire le processus de production de logiciels pour répondre à ces problématiques.

C'est autant un méta-modèle de conception de processus qu'un framework conceptuel qui met à disposition les outils et les concepts pour modéliser, documenter, présenter, gérer et rendre concret le développement. La mise en œuvre de ce méta-modèle sera généralement effectuée par un ingénieur processus, une direction projet ou un gestionnaire de programme ; plus généralement toute personne en charge de l'organisation des projets de développement ou même d'un référentiel processus au sens large.

Composant d'un cadre éprouvé

Pour expliquer le contexte de SPEM, il faut partir des travaux de l'OMG "Object Management Group". Actuellement il existe un grand nombre de méta-modèles différents qui s'inscrivent dans le cadre défini par l'OMG dont l'effort le plus significatif s'appelle MDA "Model Driven Architecture".

Cette approche regroupe les grands principes d'automatisation de la génération de composants logiciels mais elle est intéressante pour tout le système d'information. L'approche MDA pose le principe d'indépendance vis-à-vis de la plateforme d'exécution comme un élément central. En prônant un découpage du cycle de vie de la production d'un logiciel autour de PIM "Platform Independent Model" pour l'aspect métier et PSM "Platform Specific Model" pour l'aspect technique, MDA s'inscrit dans le courant voulant que la formalisation de la spécification d'un composant amène automatiquement à la génération de code.

Ainsi MDA peut être présenté comme une démarche permettant le développement de logiciels à partir de modèles et dont le paradigme serait "modéliser une fois, générer n'importe où", rappelant ainsi les travaux actuels sur les normes dans le BPM (BPMN et BPEL).

L'autre aspect qui nous intéresse ici s'appelle MOF "Meta-Object Facility". MOF est un méta-modèle qui permet de représenter des formalismes, ainsi MOF est utilisé pour décrire UML "Unified Modeling Language". MOF est donc un modèle de méta-modèle (ou méta méta-modèle) qui se situe à la base de toute définition de méta-modèle (il est donc auto-descriptif).

MOF est la concrétisation de l'approche MDA. Encore une fois les objectifs étant la réutilisation et l'interopérabilité, l'échange de modèles basés sur MOF via XMI "XML Metadata Interchange" est un exemple typique de l'utilisation qui peut en être faite.

L'OMG a donc standardisé des méta-modèles dédiés à plusieurs contextes différents : UML pour la modélisation objet, CWM "Common Warehouse Metamodel" pour la modélisation de la gestion des données et celui nous intéresse : SPEM.

Le méta-modèle SPEM, les principes

Ci-dessus, l'architecture 4 niveaux de l'OMG

Actuellement adopté dans sa version 1.1 en janvier 2005, la spécification de la version 2.0 a été rendue publique. La dernière version en date est celle du 3 mars 2007.

Après avoir présenté le contexte dans lequel s'inscrit SPEM, il est important de voir que SPEM est à la fois un méta-modèle, c'est-à-dire un langage de modélisation permettant d'exprimer les concepts, conforme au MOF et un profil UML. L'utilisation d'un profil UML fait apparaître le fait que SPEM est basé sur UML. Il est donc dérivé d'UML dont des aspects ont été spécialisés pour répondre à l'objectif de SPEM, certains concepts ont été conservés et d'autres écartés. La spécialisation par l'intermédiaire d'un profil fait ainsi passer de la modélisation « contemplative » à de la modélisation « productive ».

Le leitmotiv de SPEM est qu'un processus de développement logiciel est une collaboration entre des rôles exécutant des activités sur des produits.

SPEM 2.0 est structuré en 7 groupes principaux :

- Noyau : Le noyau contient les éléments de base (sous forme de classes UML) utilisés dans tous les autres groupes.
- Structure du processus : Ce groupe décrit les bases de création pour tous les modèles en permettant l'utilisation de modèles simples et flexibles. Il fournit des mécanismes pour les processus tels que l'application dynamique de modèle de processus à un autre processus.
- Comportement du processus : le groupe décrivant la structure du processus décrit un processus statique composé d'activités dont les dépendances sont décrites. Le groupe comportement du processus permet d'étendre ces structures avec des modèles comportementaux. Il ne définit pas pour autant sa propre approche de modélisation du

comportement mais propose des liens vers les modèles externes. Par exemple, un processus dont la structure a été décrite peut être relié à un diagramme d'activités UML 2 pour en préciser le comportement.

- Contenu géré : les processus de développement sont souvent représentés sous forme de modèles mais également sous forme de documents et de descriptions. Pour la plupart des méthodes de développement, la documentation textuelle est plus importante que la modélisation de la méthode elle-même. Dans les méthodes agiles (XP, RUP), la part belle est plutôt laissée à l'approche créative au travers de descriptions textuelles au lieu de processus formalisés. Ce groupe fournit les concepts pour gérer les descriptions textuelles soit de manière indépendante, soit en combinaison avec les concepts des structures de processus.

- Contenu de la méthode : ce groupe fournit les concepts pour que les utilisateurs de SPEM 2.0 développent une base de connaissances indépendante d'un processus ou d'un projet particulier. L'utilisateur peut donc construire une base de connaissances servant de guide sans même avoir construit un processus.

- Processus avec méthode : le groupe permet de définir ou de redéfinir les structures pour intégrer les processus conçus avec les concepts du groupe "structure de processus" avec une instance du groupe "Contenu de la méthode".

- Plugin : dans ce groupe, seront gérés les bibliothèques et les référentiels méthodologiques. L'ensemble des capacités d'un processus pourra être organisé sous forme de couches. L'utilisateur pourra ainsi enrichir ou utiliser tout ou partie des processus définis.

Concernant plus précisément les éléments servant à modéliser, la notion de WorkDefinition est centrale. Elle décrit l'activité à effectuer dans un processus et peut se décomposer en d'autres WorkDefinition. Elle est détenue par un ProcessPerformer, peut être réalisée par un ProcessRoles (Rôle décrivant un ensemble de compétences joué par une ou plusieurs personnes) et concerne un WorkProduct. La notion de WorkDefinition utilise un WorkProduct en entrée et crée ou met à jour un WorkProduct une fois traité. Un WorkProduct représente tout ce qui est produit, utilisé ou modifié par un processus.

D'autres notions existent : LifeCycle (Cycle de vie) pour définir et organiser les phases et itérations. Les notions d'Activity (Activité) et de Step (Etape) sont également disponibles. Les activités peuvent être regroupées en Disciplines (ensemble de thèmes communs).

Quel usage ?

L'un des exemples d'applications des travaux de l'OMG est le projet Eclipse Process Framework (EPF). Proposé par IBM, ce projet a pour objectif de créer un framework de base et des exemples d'outils pour l'ingénierie de processus logiciel avec comme finalité la diffusion d'exemples de processus extensibles supportant les bonnes pratiques de développement itératif.

EPF se compose d'un méta-modèle entièrement basé sur SPEM 2.0 et d'un framework de base de processus. Les méthodes et les processus sont structurés selon le méta-modèle SPEM avec des spécifications utilisant MOF, des diagrammes UML ainsi qu'un schéma XML associé.

Il existe également quelques outils de modélisation qui supportent SPEM (projet Apes) et qui permettent de valider que le processus est conforme à la norme ; cependant ces outils sont encore rares.

En conclusion

C'est par ce type de mise en œuvre que la diffusion et l'utilisation de SPEM pourront se faire tout comme l'utilisation des langages de programmation objets a propulsé UML comme langage de modélisation objet de référence. En apportant une méthode ou un processus type à suivre pour le développement logiciel, SPEM va plus loin qu'UML en ne se limitant pas à un langage de représentation. Gageons que dans l'avenir, les procédés de développement logiciels seront basés sur ce méta-modèle preuve s'il en est que la modélisation de processus peut apporter des solutions dans des domaines variés.