

Exercice FilRouge, pour que chaque étudiant joue tous les rôles au sein d'un projet.

Frédéric Baucher*¹

¹ Institut national des sciences appliquées Rouen Normandie (INSA Rouen Normandie) – INSA
– Avenue de l'Université 76801 Saint-Etienne-du-Rouvray Cedex, France

Résumé

L'exercice pédagogique FilRouge (développé à l'INSA-ROUEN et pratiqué dans un espace "LearningLab") a pour but de former au projet de développement logiciel. Cet exercice a été scénarisé notamment pour permettre à chacun de jouer les différents rôles, notamment de client (MOA¹) ou de prestataire (MOE²) afin d'appréhender les préoccupations réciproques.

L'exercice FilRouge est le résultat d'une construction itérative, sur une dizaine d'années. Les adaptations apportées au fil du temps ont permis de bâtir un modèle avec des invariants et des points de flexibilité permettant d'intégrer les nouveaux outils, pratiques, ... qui ne manquent pas d'apparaître (agilité, ...).

L'objet de la communication est de proposer une généralisation du modèle de cet exercice pour une éventuelle application à la pratique de projets sur des domaines autres que le développement logiciel.

Mots-clés : pédagogie par projet, génie logiciel, situations de collaboration, learninglab

1. Motivation et processus retenu

Tout enseignement qui choisit d'aborder la notion de processus, notamment par la pratique est confronté à ces constats, illustrés dans le domaine logiciel :

- un processus de développement logiciel, exécuté dans une organisation **en milieu économique** a pour but de produire une application logicielle de qualité au meilleur coût ;

* Professeur Associé

¹ MOA : **M**aitrise d'**O**uvr**A**ge, même si ces termes (MOA et MOE) commence à être obsolète, il perdure et on trouve des équivalents dans les milieux agiles. Pour MOA, *Product Owner* est un équivalent possible (mais il désigne une personne et non un individu).

² MOE : **M**aitrise d'**O**Ev**u**vre. Pour MOE, *Scrum Master* est un équivalent possible (mais il désigne une personne et non un individu).

- en revanche, pour un processus exécuté par des étudiants **en milieu académique**, la finalité est avant tout de former l'étudiant.

Même si les finalités sont différentes, les avancées en matière de formalisation des processus observées depuis les appels³ à théoriser le Génie Logiciel ont contribué aussi à étoffer l'enseignement de cette discipline.

L'exercice FilRouge s'inscrit dans cette démarche et il a été conçu pour l'enseignement du formalisme UML mais aussi des méthodes applicables indépendamment des modes et tendances. Cet exercice s'est appuyé sur un processus de développement logiciel ad'hoc, MiniP, dont une des premières versions est évoquée dans [2][ECHCHADI&'2007].

L'exercice FilRouge (et le processus MiniP) ont pour objectif de répondre aux constatations suivantes :

- la discipline (Génie Logiciel) demande la production de nombreux livrables intermédiaires. La pertinence et le sens de ces derniers ne peuvent apparaître que dans une mise en contexte au travers d'un processus global :
 - en partant de l'intuition initiale
 - puis en passant par **le cahier des charges**
 - et enfin en clôturant avec le livrable final, à savoir l'application ou le service en version 1.
- les diagrammes UML (et la plupart des livrables documentaires) ont pour principale fonction d'assurer la transmission d'une vue du domaine étudié d'un acteur (ou rôle) à un autre. L'objectif est de sensibiliser tout particulièrement les étudiants à la difficulté de communiquer les exigences sans pertes et/ou incompréhensions du client au prestataire. Dans ce but, les activités doivent placer successivement l'étudiant dans des situations de producteur puis de consommateur des artefacts documentaires. Le processus fait donc intervenir deux équipes : l'une représentant le client ou MOA (qui travaille principalement jusqu'à la production du cahier des charges), l'autre le prestataire ou MOE (qui travaille activement à partir de la production du cahier des charges). Pour une égalité de traitement des étudiants, il est nécessaire **de faire jouer à chacun les rôles relatifs aux organisations cliente et prestataire**.

2. Quelques contingences propres au milieu pédagogique

La littérature consacrée au processus de développement logiciel se concentre principalement sur des configurations adaptées à des organisations évoluant en **milieu économique**. La transposition en **milieu académique** d'un processus doit attirer sur plusieurs points d'attention, illustrés dans la présente section par des exemples concrets. Avant d'étudier, dans le chapitre suivant, l'abstraction du monde réel académique, observons dans la table I quelques uns des objets de ce monde réel, en l'occurrence, celui du département ASI de l'INSA Rouen Normandie.

³ cf des articles fondateurs comme [JACOBSON'2009].

TABLE I. ELÉMENTS (OBJETS) DU MONDE RÉEL ACADEMIQUE

- la promotion de 54 étudiants (Kevin, Manon, ...) inscrits dans un module de Génie Logiciel (EC UMLP) de l'année 2019,
- les différentes séances au cours d'un semestre (ie : séance du 08/03/2019 du semestre de Printemps 2019)
- l'organisation académique en 2 groupes de TD (qui subdivise la promotion en des groupe de +/- 24 étudiants),
- l'organisation pédagogique dans la promotion en 16 équipes (1a, 1b, ..., 2k, 2m, ...) de 3 à 4 étudiants, à raison de 8 équipes par **groupe** de TD.

2.1. Egalité de traitement : la pirouette MOA/MOE

Dans chaque groupe de TD, des équipes sont formées. Au démarrage du processus, ces équipes joue le rôle de MOA sur une intuition de produit/service (voir exemples sur la table II). Cette intuition est propre à une équipe qui applique, dans un rôle de MOA, les différentes activités du processus jusqu'à produire un cahier des charges. A l'issue de cette première phase, chaque équipe choisit ensuite, selon le principe « premier arrivé, premier servi », le cahier des charges d'une autre équipe, sur lequel elle jouera le rôle de MOE pour mettre en œuvre les exigences qui y sont consignées. Ce dispositif (appelé la pirouette MOA/MOE) permet ainsi à chaque équipe d'expérimenter le rôle de MOA et MOE, sur des projets respectifs.

TABLE II. EXEMPLES DE PROJET PROPOSÉS PAR DES ÉQUIPES EN TANT QUE MOA

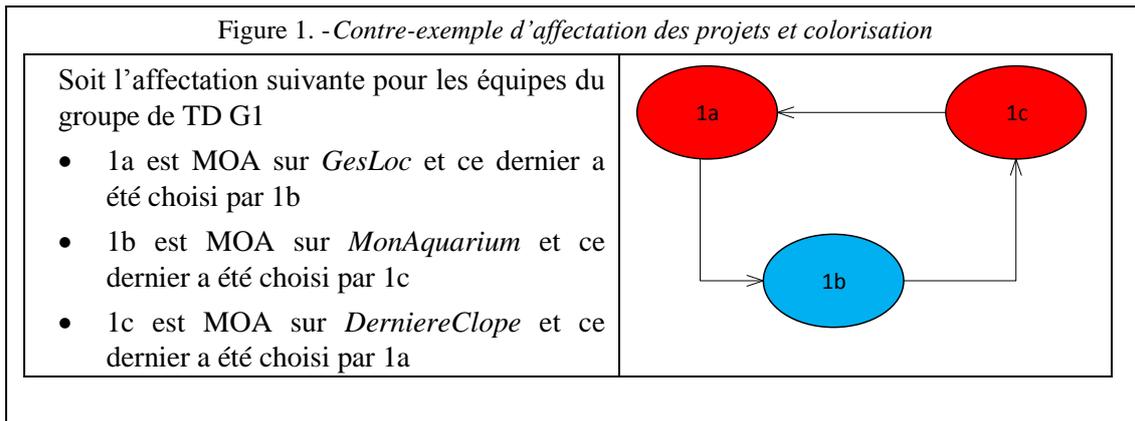
Chaque équipe (1a, 1b, ...) travaille en tant que MOE sur une intuition :	
Equipe 1a => <i>GesLoc (une gestion de co-location étudiant)</i>	
Equipe 1b => <i>MonAquarium (optimiser l'entretien d'un parc d'aquarium)</i>	
Equipe 1c => <i>DerniereClope (accompagner un fumeur dans son sevrage)</i>	

Pour rendre possible cet exercice dans un contexte pédagogique, les équipes sont séparées **en deux partitions** : les rouges et les bleues. En effet, certaines activités nécessitent de réunir les deux équipes (MOA et MOE) sur le même projet sur un même temps. La séance de TD relative à cet exercice est donc divisée en deux temps :

- Temps 1 : les équipes rouges jouent le rôle de MOA, les équipes bleues jouent le rôle de MOE
- Temps 2 : les équipes bleues jouent le rôle de MOA, les équipes rouges jouent le rôle de MOE

Cette organisation impose cependant une contrainte, dont un contre-exemple est illustré sur la figure 1. Cette configuration n'est pas conformes car les équipes 1a et 1c seront mobilisées dans le même temps pour leur rôle MOA => elles ne peuvent se rencontrer.

Figure 1. - Contre-exemple d'affectation des projets et colorisation



Pour s'assurer de la conformité de l'affectation avec l'organisation des TD, une représentation sous forme de graphe orienté a été adoptée :

- Chaque équipe est un sommet,
- Le choix du sujet d'une équipe MOA (ie GestLoc) par une équipe MOE est illustré par une arête orienté du sommet MOE vers le sommet MOE.

Pour vérifier qu'un ensemble d'affectations sur un groupe de TD est conforme, il faut s'assurer que le graphe est **2-coloriable** (on dit aussi **biparti**), c'est-à-dire :

- Extrait de [3][TOPART'2011] en page 11 : *Un graphe est biparti lorsque l'ensemble des sommets V peut être partitionné en deux ensembles V1 (ie bleu) et V2 (ie rouge) tels que $uv \in E$ implique $u \in V1$ et $v \in V2$ (où uv signifie que le sommet u possède une arête avec le sommet v).*

Sachant que le choix des sujets est contraint par le principe du premier arrivé, premier servi, des problèmes d'affectations ne peuvent se produire qu'à partir de 6 équipes.

En conclusion, le dispositif de pirouette MOA/MOE est possible selon les contraintes suivantes :

- Nombre paire d'équipes dans un groupe de TD ;
- Les choix de cahier des charges (dans le cadre de la pirouette MOA/MOE) doit produire un graphe biparti (quitte à modifier les choix de certaines équipes pour respecter cette contrainte).

2.2. Nécessité d'un LearningLab

Les activités en groupe de TD de l'exercice FilRouge donnent lieu à de nombreuses situations de collaboration :

- soit au sein d'une équipe (ie : activité « élaborer les fiches personae »),
- soit entre deux équipes, à raison d'une MOE et d'une MOA (ie : activité : « mener réunion de lancement »).

Ces différentes situations de collaboration nécessitent le recours à différents moyens, comme illustré de manière non exhaustive dans la table III.

TABLE III. QUELQUES SITUATIONS DE COLLABORATION ET LES MOYENS NECESSAIRES

Activités	Nombre d'équipes	Tables	Mur d'écriture	Tableau blanc	Ecran ⁴
Mener le remue-méninge	2 en local	N/A	X=1 Y	Y=1 X	0
Elaborer les fiches personae	1 en local	4 places	0	0 ou 1	0
Mener la réunion de lancement	2 en local	8 places	0	X=1 Y	Y=1 X

Notation : **X=1 | Y** : la présente colonne doit valoir 1, si la colonne Y vaut 0.

Seule une salle de type LearningLab (cf [4][LEARNINGLAB'2014]⁵) offre les moyens et la mobilité capables de répondre à ces différentes configurations. Dans la mesure où les moyens sont limités (2 murs d'écritures disponibles) alors que certains TD peuvent accueillir jusqu'à 4 situation de collaboration nécessitant ce moyen, des alternatives sont proposées (ie pour l'activité « Mener le remue-méninge », chacune des situations peut être réalisée soit sur un murs d'écriture, soit sur un tableau blanc).

TABLE IV. LES MOYENS DISPONIBLES DANS LA SALLE LEARNINGLAB À ROUEN

Activités	Q ^{ité}
Tables amovibles	16
Sièges amovibles	32
Ecrans amovibles	4
Tableaux blancs amovibles	4
Murs d'écriture fixes	2
Vidéo-projecteurs fixes	2

3. Nouvelles exigences et recours à un méta-modèle

L'exercice FilRouge, dans ses premières versions, reposait sur un processus **monolithique** (un seul chemin possible) applicable à **une seule promotion**, à un **niveau donné** (L3) avec un **ensemble figé d'éléments** (principalement livrables et activités). Une représentation par un diagramme d'activité (au sens UML) et un tableau des livraisons suffisait alors.

De nouvelles exigences sont apparues :

- Le processus doit être applicable à différents niveaux (ie : L3, M1)
- Une application simultanée à plusieurs promotions dans des établissements différents (ie : L3 à INSA Rouen Normandie et L2 à EUROMED)
- La possibilité de chemins différents en fonction du contexte du projet d'une équipe.

⁴ La contrainte est que l'écran doit pouvoir être près de la table de travail des étudiants.

⁵ L'INSA Rouen Normandie est membre du réseau *LearningLab Network* et dispose, à ce titre, d'une seule répondant à ce cahier des charges.

- Les documentations des outils applicables sur certaines activités doivent être adaptées selon le contexte d'utilisation (pour la gestion des sources : *gitlab* local si disponible, ou *github.com* en l'absence d'un local).

De plus, certains points nécessitent des améliorations :

- Des étudiants avaient des difficultés à appréhender les différents types de supports documentant le processus (ie : mode opératoire d'une activité, formalisme dans lequel fournir un livrable)

En fonction des contextes d'utilisation du processus, même si une grande partie est commune, des points de flexibilité apparaissent. La duplication du processus initiale en autant de copies que de contextes d'utilisation présente rapidement ses limites, notamment dans le cas de la mise à jour d'une partie commune à chacun des processus.

Deux exigences sont alors apparues :

- pouvoir distinguer, d'une part, les éléments de méthodes réutilisables et, d'autre part, les processus qui incluent ces éléments de méthodes ;
- mettre en place un modèle de processus, abstraction des processus concrets (on parlera aussi de plan-projet) appliqués sur un projet donné par une équipe donnée. Ce modèle doit être capable d'encapsuler les invariants et les propriétés communes, ces dernières pouvant être valorisées au moment de l'instanciation du processus dans un plan-projet.

Une veille sur les solutions méthodologiques applicables a rapidement mené vers la norme SPEM 2.0 [5][OMG'2008], d'autant que cette dernière est souvent présentée aux côtés de UML sur les diagrammes illustrant les normes de l'OMG (telle la figure 2).

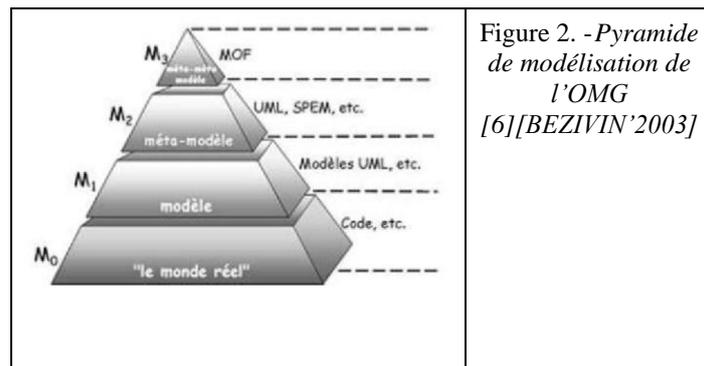


Figure 2. -Pyramide de modélisation de l'OMG [6][BEZIVIN'2003]

L'application de ce méta-modèle à notre problématique s'articule comme suit :

- M2 : norme SPEM, mise au point par l'OMG, réutilisée dans le cadre de la modélisation des processus au niveau M1 ;
- M1 : modèle (générique) du processus auquel doit être conforme tout plan projet mis en œuvre au niveau M0 ;
- M0 : plan-projet mettant en application un processus en figeant la valeur de chaque paramètre et point de flexibilité offert par le processus au niveau M1.

Le méta-modèle M2 (SPEM) repose sur une dichotomie claire entre les éléments de méthodes (niveau M1), **réutilisables**, et les processus (niveau M1 aussi), comme **enchaînements des éléments de méthodes**. Ce sont seulement ces derniers (processus) qui pourront être instancié en **des plans-projet** (niveau M0).

Cette dichotomie méthodes / processus apparait sur la figure 3 à travers les packages et leur contenu :

- **MethodContent** : *RoleDefinition, WorkProductDefinition, WorkProductKind (Artifact, Deliverable, Outcome), TaskDefinition* (cf figure 4), ... ;
- **ProcessWithMethods** : *Activity, MethodContentUse, RoleUse, WorkProductUse, TaskUse, Milestone, ...*

- **ManagedContent** : *Guidance* (un élément de description qui vient en support d'éléments issus aussi bien de **MethodContent** (ie : *RoleDefinition*, ...) que de **ProcessWithMethods** (*Activity*, ...).

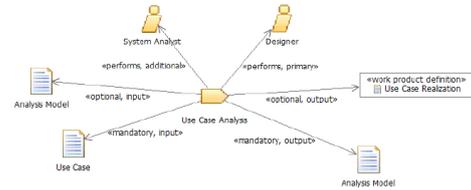
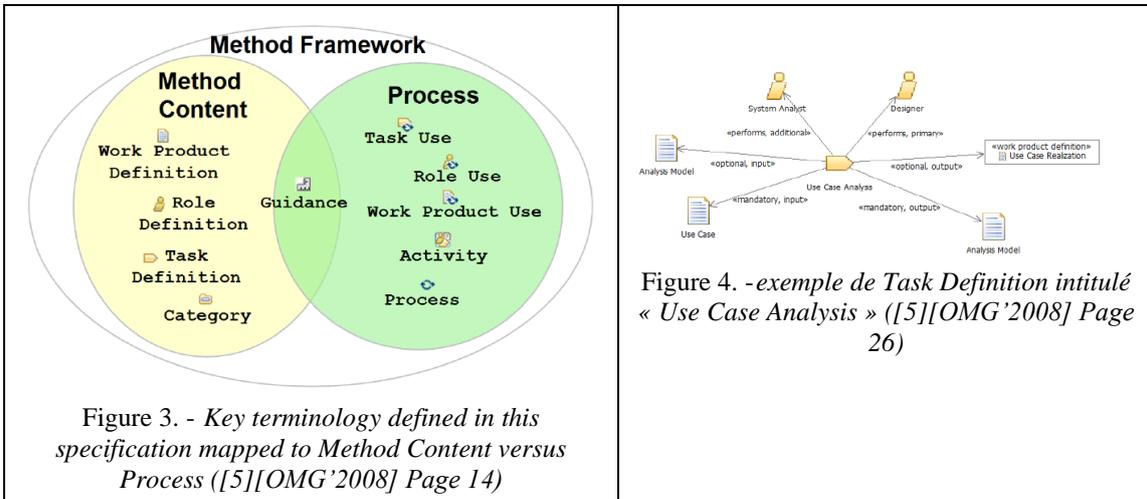
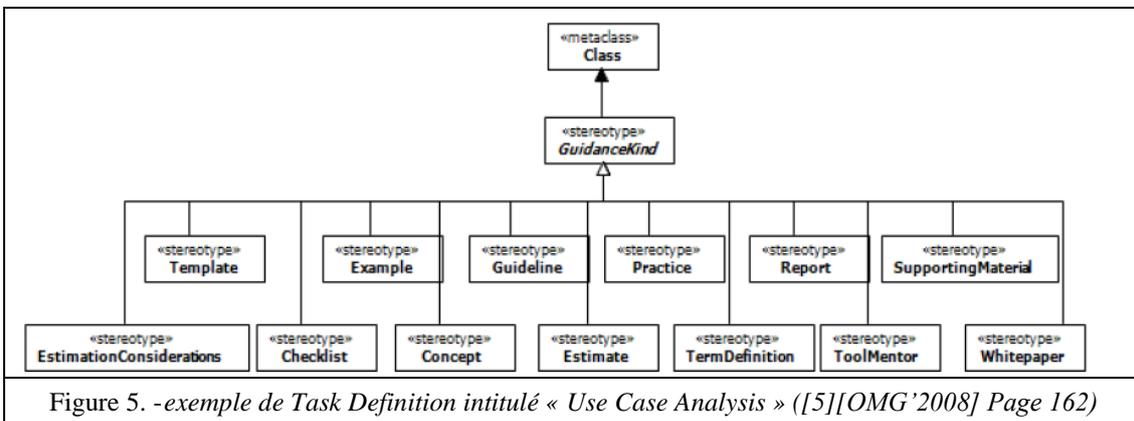


Figure 4. - exemple de Task Definition intitulé « Use Case Analysis » ([5][OMG'2008] Page 26)

3.1. Le modèle de processus comme outil de conception pédagogique

La disponibilité d'un méta-modèle (niveau M2) et de sa documentation présente de nombreuses opportunités pour outiller une famille de processus dans un modèle générique de processus (niveau M1).

A titre d'exemple, pour répondre notamment à la difficulté des étudiants à appréhender les différents types de documentation (*Guidance* dans la parlance SPEM), le méta-modèle prévoit en effet de nombreux types (voir figure 5).



Ces types de *Guidance* sont précisément définis, on citera notamment :

- **Example** : represents a typical, partially completed, sample instance of one or more work products or scenario-like description of how Task may be performed ([5][OMG'2008] p163)
- **Templates** : provides for a work product a predefined table of contents, sections, packages, and/or headings, a standardized format ([5][OMG'2008] p164)
- **ToolMentor** : shows how to use a specific tool to accomplish some piece of work either in the context of or independent from a Task (MethodContent) or Activity (ProcessWithMethods) ([5][OMG'2008] p165)

Cependant, des lacunes subsistent si l'on souhaite modéliser la documentation disponible dans un contexte pédagogique : ainsi un **exercice** peut être considéré comme un *Example* mais ce dernier doit être étendu avec deux composants supplémentaires, à savoir **l'énoncé et la solution**. Il s'agit d'un exemple d'éléments qu'il serait nécessaire d'ajouter dans le cadre d'une démarche d'extension du modèle SPEM dans un profil orienté « Academic », si cette dernière est retenue (voir conclusion).

3.2. Transformation d'un processus économique en un processus pédagogique

En termes de conception de processus, la démarche recommandée peut se scinder en deux phases :

- Une phase de conception, en l'absence de toute contrainte académique, qui aboutit à un processus « fonctionnel » (qui pourrait éventuellement être appliqué en milieu économique).
- Une phase de transformation de ce dernier en un processus adapté à un contexte académique (on parlera de processus pédagogique).

Au-delà de l'intérêt didactique du modèle (comme illustré précédemment), ce dernier apporte un atout opérationnel au cours de la phase de transformation.

A titre d'illustration de l'apport d'un modèle dans cette phase, une des bonnes pratiques, à savoir l'adaptation aux contraintes de temps, est décrite dans la suite.

En effet, le temps à consacrer à un enseignement, et par suite à un exercice (comme FilRouge), est encadré par la maquette pédagogique. A ce titre, les temps des séances (de TD ou de TP) aussi bien que le travail personnel est limité.

C'est pourquoi une attention toute particulière doit être portée sur les activités répétables⁶ (ou activité itérative) ou à occurrences multiples⁷ (ou activité incrémentale). Ces caractéristiques sont renseignées dans le modèle et peuvent donc être exploitées lors de la transformation du modèle de processus « fonctionnel » en un modèle de processus pédagogique. L'objectif sur ces activités est de cadrer précisément le nombre exigé dans le cadre de l'exercice en fixant le nombre d'exécution de l'activité suffisant pour atteindre l'objectif pédagogique ;

Ainsi, la tâche « Use Case Analysis » (qui est à occurrence multiple) est à répéter autant de fois qu'il y a de cas d'utilisation. Dans un cadre pédagogique, on se limitera à trois cas, en exigeant que chacun de ces cas mettent en œuvre un des types de documentation d'enchaînement des actions (Tableau de Cockburn, DSS ou diagramme d'activité).

Une bonne pratique consiste :

- à ajouter une étape de priorisation en amont qui permet de limiter au nombre souhaité, pour une activité à occurrence multiple ;
- à expliciter très clairement les critères de finition d'un livrable sur une activité répétable, afin que l'étudiant régule le nombre d'itération en fonction de ces critères.

3.3. Mise en œuvre d'un processus pédagogique en plan projet

Une fois la transformation en processus pédagogique terminée, ce dernier peut être mis en œuvre⁸ par chacune des équipes d'une promotion. Cette instanciation va permettre d'indiquer la valeur de chacun des paramètres du processus dans un plan-projet.

⁶ attribut *isRepeatable* sur *BreakdownElement* dans le package *ProcessWithMethods*.

⁷ attribut *hasMultipleOccurrences* sur *WorkBreakdownElement* dans le package *ProcessWithMethods*.

⁸ On parle de *enactment* dans la norme SPEM

A titre d'exemple, pour citer les paramètres les plus intuitifs : le nom de l'équipe, le nom des étudiants, les jours des jalons de remises des livrables (instance d'un *milestone*), les horaires des séances, ...

4. Conclusion

Après avoir étudié quelques contraintes liées à la scénarisation du déroulement d'un processus de développement logiciel dans un contexte pédagogique, l'utilisation d'un méta-modèle comme SPEM 2.0 a été illustré à travers quelques exemples.

Plusieurs pistes d'évolutions sont à explorer :

- Le processus proposé aux étudiants est figé, aux paramètres d'information à renseigner près. Une évolution possible serait de laisser la liberté à l'étudiant (en M2 ou M3) d'expérimenter son propre processus en l'autorisant à activer certaines activités. Le système devrait être alors en capacité de fournir vérification quant à la conformité du processus proposé avec les règles embarquée dans le méta-modèle.
- Même si l'apport de SPEM est précieux pour l'exercice FilRouge, il est apparu que SPEM, dans sa version actuelle, souffrait quelques limitations pour une application en contexte pédagogique. C'est pourquoi deux pistes sont explorés pour la suite :
 - La définition d'un profil Education qui étendrait le méta-modèle SPEM,
 - La migration vers une autre solution méthodologique ESSENCE (cf [7][OMG'2018]) plus récent, qui se définit non plus comme méta-modèle mais noyau et langage.

Enfin, une dernière piste d'évolution consisterait à étudier l'adaptation de l'exercice FilRouge à d'autres domaines que le génie logiciel.

Références (style TitreSansNumero)

- [1] [JACOBSON'2009] Ivar Jacobson, Bertrand Meyer (2009, Aout, 06), Methods Need Theory, Dr Dobbs Journal. Disponible sur <http://www.drdoobbs.com/architecture-and-design/methods-need-theory/219100242#>. Consulté le 31/03/2019.
- [2] [ECHCHADI'2007] Said Echchadi, Frédéric_Baucher, Brahim_Erraha (2007, Juillet). *MiniP : a development process for software engineering students in multi cultural environment*. Poster présenté à IEEE Global Information Infrastructure Symposium – Marrakech, MA.
- [3] [TOPART'2007] Hélène Topart. Etude d'une nouvelle classe de graphes : les graphes hypotriangulés. Réseaux et télécommunications [cs.NI]. Conservatoire national des arts et metiers - CNAM, 2011.
- [4] [LEARNINGLAB'2014] LearningLab Network, Charte du LearningLab Network, [en ligne] 2014. Disponible sur <http://www.learninglab-network.com/wp-content/uploads/2014/12/Charte-Learning-Lab-Network-.pdf>. Consulté le 31/03/2019.
- [5] [OMG'2008] OMG (2008), Software & Systems Process Engineering Meta-Model Specification Version 2.0, Disponible sur <http://www.omg.org/spec/SPEM/2.0/PDF>. Consulté le 31/03/2019.

- [6] [BEZIVIN'2003] Bézivin J., La transformation de modèles, Ecole d'Eté d'Informatique CEA EDF INRIA [cours], INRIA-ATLAS & Université de Nantes, 2003.
- [7] [OMG'2016] OMG (2016), Essence – Kernel and Language for Software Engineering Methods Version 1.2, Disponible sur <https://www.omg.org/spec/Essence/1.2/PDF>. Consulté le 31/03/2019.