Adaptive Guidance System for SPEM

Hamid Khemissa[1] Mohamed Ahmed-nacer[2] Mourad Oussalah[3]

[1&2] Computer Systems Laboratory, Computer Science Institute, USTHB
University, El Alia BP n° 32, Bab Ezzouar Algeria. Tel/Fax (00)213 21247917
[1]`hkhemissa@gmail.com`,   [2]`anacer@cerist.dz`
[3]Computer Laboratory Nantes Atlantique, Faculty of science, Nantes
University
2 rue de la Houssiniere- BP 92208 44322 nantes Cedex 3. Tel 0251125847,
Fax 0251125858
`Mourad.Oussalah@univ-nantes.fr`

ABSTRACT

This paper deals with an adaptive guidance system for software processes. The aim is to define an approach of process modeling with a recursive adaptation of the guidance. This one is specifically adapted to the developer model through its role and its qualification, and in relation to the activity model associated with the context of the activity in progress. This guidance system is provided on the basis on the choice criteria like the suitable access mode, the assistance function to be ensured and the object of assistance to be considered.

The description of the suggested approach is done using the modeling formalism SPEM extended by new concepts and principles dedicated to the modeling of the adaptive guidance.

KEYWORDS

Meta Modeling, Software Process Modeling, Guidance Profile, Adaptive Guidance, SPEM.Extension

1. INTRODUCTION

To attempt a better software quality and to keep a consistency and productivity aspects in software development, it requires assisting developers at both organizational and methodological levels as well as on the software product consistency [1]. A rigorous assistance targets two aspects: 1) The control progress of the software process development regarding the temporal constraints of activity and the consistency of the results, and 2) the guidance adapted to the specific needs within the context of the activity in progress.

A guidance model to the software engineering should gather the essential characteristics making it possible to build the assistance system and to graft it with the development system. Several process-centered environments [2] dealing with the assistance aspect in the support of the software product development. The runtime software of these environments guides and supports the user in application of the software process activities.

The principle of the proposed approach is to consider that a process of software development is entirely based on collaboration between actors, who perform activities to take in a common goal called software product. Each actor supports the achievement of a particular task. This work

requires a refined guidance and adapted to the actor's profile. This profile is described respectively by the developer model (defined by its role, qualification and behavior) and the activity model, in relation to the context of the task in progress. The guidance system must be able to agree to the activity in all its complexity and to provide recursive adaptation related to the task progression.

Our approach uses SPEM [6] (Software Process Engineering and system Meta model) that is considered as the most used Meta model providing basic concepts necessary to model various software processes. So, the proposed approach gives definition concepts and principles of adaptive guidance, that are extended to SPEM in form of stereotypes relating to guidance in the packages' profile "MethodContent" and "ProcessWithsMethod" as well as their instantiation, respectively, by new stereotyped classes and associations.

Section 2 of this paper presents a synthesis of similar work and describes the current trend. Section 3 presents our approach of the guidance adaptive modeling while section 4 describes the new guidance concepts we have integrated in the Meta Model SPEM. Section 5 presents the cases study of the adaptive guidance. It ends with conclusion and future prospects.

2. CURRENT TENDS

Several process-centered environments [2, 7, 8, 9] deal with the assistance aspect in the support of the software product development. Some PSEEs use an assistance description structured in phases like prescribing systems or proactive systems to control the operations carried out by the developer. The principal limitations of these PSEEs are:

- The human actor has a central role in the progress of the development process regardless of his profile (qualifications and behavior).
- The basic guidance is defined as a global orientation core whatever the profile of both the activity and the developer.
- The selection of the appropriate type of guidance is more often intuitive and not suitable.

To respond to this limit, one currently tries to offer more flexibility in the language of process software modeling. This tendency results in the idea to define interventions of direct and adaptive assistance in particular contexts during the software process progress. Among this new generation of following PSEEs, the M1 level of meta-model SPEM is included:

ADELE/APEL [10] is based on a reactive database. It proposes a global assistance of proscriptive type and automates part of the development process using triggers.

RHODES/PBOOL+ [11, 17] uses an explicit description of a development process. The software processes are modeled in PBOOL language. The activities are associated to a guidance system with various scenarios of possible realization.

ADDD/ALADYN [2] provides process automation and control the impact in a concrete system. The task hierarchy is used to organize descriptions of process, called policies. Several aspects are grouped and treated in a policy. A policy can be instantiated for several tasks. The instantiated triggers are rules of the form event-condition-action (ECA) and used to implement a reactive behavior.

On the M2 level of Meta model, SPEM [6] introduced the concept of "Guidance" in the "*Managed Content*" package by defining the stereotype "Guidance" (**see Figure-1-**). According to SPEM," the Guidance is a describable element which provides additional information to define

the describable elements of a modeling. It also offers, through the stereotype "Guidance_kind" different types of guidance such as: Template, Guidelines, Checklists, etc. ...

However, the selection of guidance types remains defined in a manual and in an intuitive way. It depends on the experience and on the informal personality of the project manager. In addition, the proposed guidance is not suitable to the actor's profile (role, qualifications and behavior)
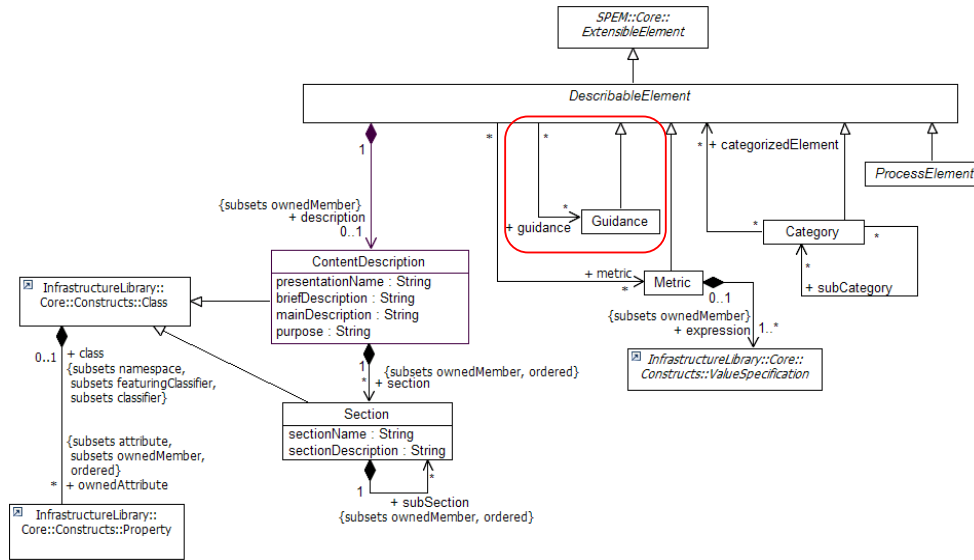

Figure-1- : "ManagedContent" package.

Except at the package level "ManagedContent", SPEM does not develop the guidance concept. Its definition and practice across the various packages is not approached in details. Thus, SPEM does not offer models of preset guidance nor directives of uses and selection of Guidances_Kind.

Nevertheless, the "Method Content" package defines the notion of qualification (see Figure-2-). The qualification describes the skill provided by a role or qualifications required for a task. It should be noted that the qualification mentioned during the definition of the task and the role is not expressed at the package level "ProcessWithMethod".
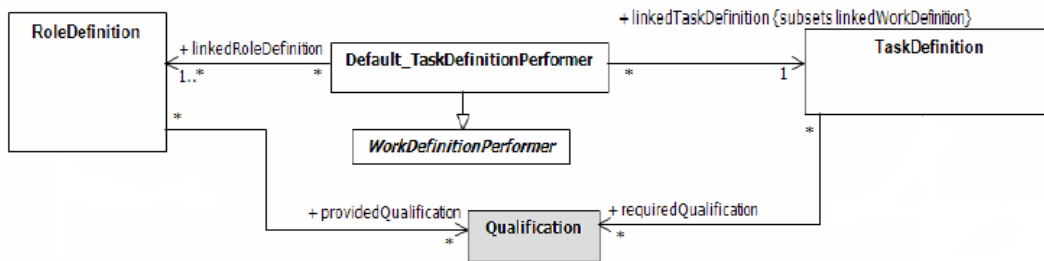

Figure-2- : The qualification concept in the « MethodContent » package.

A comparative table of the studied PSEEs and SPEM is described as follows:

Environment \\ Characteristics	ADELE/APEL	RHODES / PBOOL+	ADDD / ALADIN	SPEM	AGM (**Adaptive Guidance Model**)
Global guidance core	Global	Global	Customized for each task	Global	Customized for each task
Human actor profile oriented guidance	Not adapted	considered strategy Model	Not adapted	Not adapted	Adapted to developer's model
Context development Guidance	Not adapted	Adapted	Adapted	Not adapted	Adapted
Explicit activity abstraction	Explicit abstraction	Explicit abstraction	Implicit abstraction	Explicit abstraction	Explicit abstraction
Explicit task abstraction	Implicit abstraction	Not invoked	Explicit abstraction	Explicit abstraction	Explicit abstraction
Process Modeling Language(PML)	APEL	PBOLL+	ALADYN	UML Profile	UML Profile

Table-1: A comparative table.

AGM is defined in considering the principal limitations of PSEEs and essential characteristics of our approach as the context adaptation aspect and the abstraction levels considered.

The current tendency is that developers would like to have integrated environments that are suitable to specific needs according to the role and the characteristics of each developer. However, the provided efforts to develop such environments remain an insufficient contribution. In this context, our conceptual model is based on the typical reasoning of software processes enriched by the Adaptive Guidance element. This one controls the good progress of activities and offers an adaptive support to the actor. It is described schematically as follow:
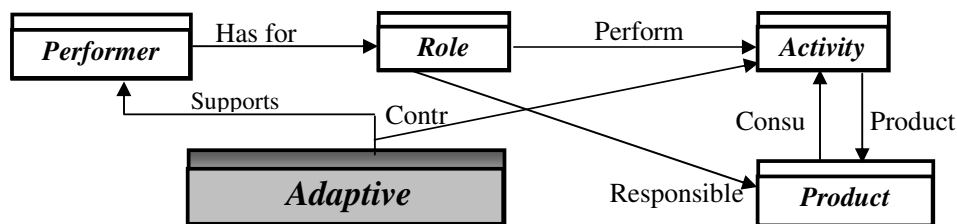

Figure-3- : Conceptual model with adaptive guidance.

This generation of guidance environment still interest researchers in defining new concepts and objectives of the software process modeling among the defined objectives [3, 4, 5, 18].

Our work proposes an approach to define adapted guidance systems. The information provided must be adapted to the development context. They must guide the actor during the software process development through suitable actions and decisions to undertake with corrective, constructive or specific intervention [5, 14].

3. THE ADAPTIVE GUIDANCE MODEL

To define an adaptive guidance model to the context, our approach considers collaboration and development levels. It defines an assistance system based on the models of the activity and the task related respectively by the process role to the corresponding models of the team and the developer. To ensure consistency between these different components, the models of developer and task inherit respectively the characteristics of the models of team and activity. This makes approach to generate the assistance interventions adapted to the development context relating with the considered models (**See Figure-4-**
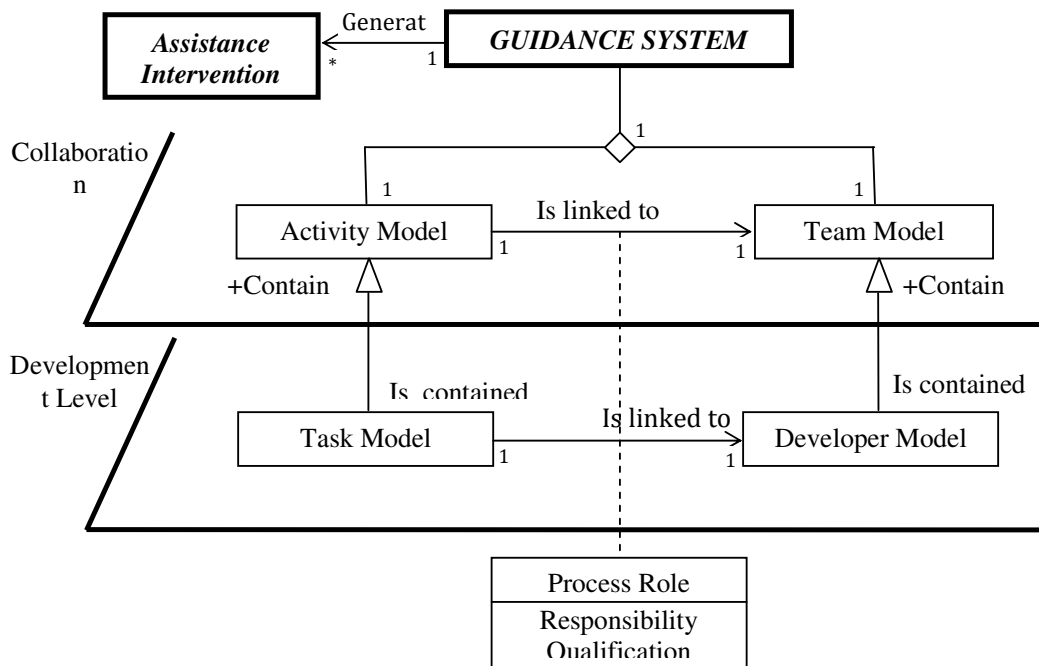
Figure-4- : The adaptive guidance model.

3.1 The main models

This assistance system is based on the major models: the activity model, the developer model and the team development model.

3.1.1 The activity model:

models the workflow, it is defined by:

▶ A hierarchical list of tasks,

► A mode of progression in the activity ensuring that all tasks can be performed under control in a preset order established by the designer,
► A temporal mode of progression specifying deadlines for completion.

The aspects of the activity model are useful for the system of assistance to provide assistance on contextual growth in activity.

3.1.2 The developer model: defines the specific properties of each user. It allows our model to make adaptation according to these properties while maintaining the activity model. These properties can be either static or dynamic.

► The static aspect refers to the user characteristics:

- o his expertise in the field,
- o his familiarity with the software process model or with the software process,
- o his role in the activity.

► The dynamic aspect refers to the behavior of using the assistance system, the assistance system must be interpreted during use of the process or system software support, for example:

- o the fact to execute, to define or to complete resource of software process,
- o the workload of an activity,
- o his reaction to a message of support.

3.1.3 The development team model: development environments allow exchanges and collaborative work. The assistance system can then construct a development team model that represents elements of the team. Example: trace of the various activities of the team as well as different interactions allow the developer to have a script about his own progress in the activity and the progression of the team. The properties of this model can be static or dynamic order.

► The static dimension referencing skills and team performance in the field of collaboration and distribution of task.
► The dynamic dimension deals with the behavior of the development team. It describes the actions taken by the team during the course of software process.

(These data constitute indication that can be interpreted on the use of the assistance by the developer.)

3.2 The assistance intervention

During the construction or interpretation of a software process model, the model proposed for assistance allows the developer to choose various support functions, namely:

3.2.1 Control and taking corrective initiative: protect the user of its own initiatives when they are inappropriate, inadequate initiative under progress.

3.2.2 Control and taking constructive initiative: the ability to take positive initiatives, executing and combining the performance of operations without user intervention.

3.2.2   Specific assistance: analyze the impact projection to avoid deadlocks or delays.

3.3 The assistance access mode

Assistance is made according to an access mode, relatively to a particular situation. Our assistance system allows the following access modes:

3.3.1 Contextual assistance: intervention is provided dynamically according to the context and the state of the process. Intervention doesn't consider the developer model. For instance, to avoid inconsistency during the affectation of a resource.

3.3.2 Global adaptive assistance: intervention is provided according to the developer's static model. For instance, the user asks for explanations on his choice.

3.3.3 Contextual adaptive assistance: intervention is provided according to the context while considering the developer's dynamic model. For instance, to help the developer on the chaining principle during the software process progression.

4. ADAPTIVE GUIDANCE VIA SPEM

Respecting the synoptic elaboration of SPEM [6], we extend it with the adaptive guidance concepts in the packages level: "*MethodContent*" and "*ProcessWithMethod*". So, we introduce and illustrate in details the description of the main stereotypes "GuidanceDefinition" and "*GuidanceUse*" respectively in the package level "*MethodContent*" and *ProcessWithMethod*" in order to allow the exploitation of predefined guidance and describes each time the used guidance according to the current context.

4.1. The "MethodContent" package with adaptive guidance

The "MethodContext" package defines the basic element of each method such as Role, Task and Work Product. Most classes and stereotypes of this package have the particularity of having the suffix "Definition" in order to express the definition of the element. Respecting this logic to extend the profile of the package, we have defined the new stereotype "GuidanceDefinition" that defines any guidance which can be used in a development approach (**see Figure -5-**).
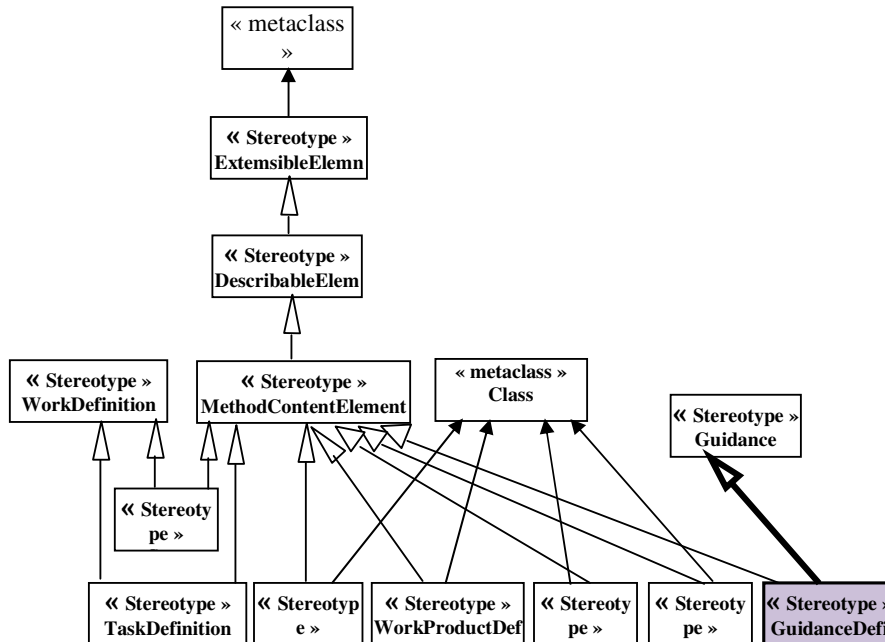

Figure-5-: Extending SPEM2.0 profile with the "GuidanceDefinition" stereotype.

The stereotyped class "GuidanceDefinition" in relation with the basic elements and their relations with the subclasses of package elements "MethodContent" are given in details in a diagram (**see Figure-6-**), and by the description of its association and its semantic definition. The relations are defined to describe how to perform the method according to the modeler perception.
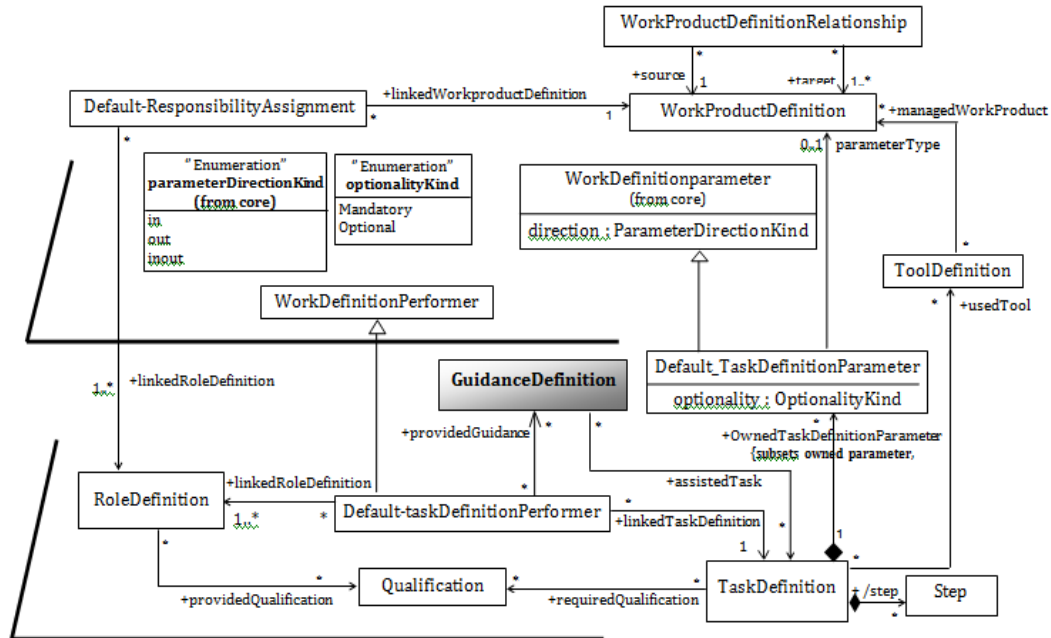

Figure-6-: Key elements of the "MethodContent » description.

► **Description of the "GuidanceDefinition" class.**
   The "GuidanceDefinition" semantic and its different associations are as follows.

a) Semantic definition

The "GuidanceDefinition" instance expresses that any guidance definition is made according to the task instant context to realize taking into consideration the developer profile. A "GuidanceDefinition" document represents the guidance offered by the instances of a role definition and / or the required guidance for the execution of a task. The definition of the actor's role and the context of a task are presumed to be useful to find the most adapted guidance during the interpretation of the predefined method content and dynamically assign the most appropriate guidance.

b) Associations

Super class: Guidance, MethodContentElement.
The "GuidanceDefinition" is described as an element of "MethodContent" that defines known and recognized guidance in the software development domain. It describes the appropriate guidance to a task definition instance taking into account the provided competences by the class default task definitions performer associated to a defined role.

   o **ProvidedGuidance : GuidanceDefinition;** this association expresses the fact, in one hand, that "*GuidanceDefinition*" provides a lot of defined guidances compared with the

characteristics of the task developer "*Default-TaskDefinitionPerformer* ". In another hand, this last may have zero or many guidance definition as an adapted support of the task developer profile.

o **Assisted task: Task Definition:** the "TaskDefiniton" instance may be assisted by zero to many adapted guidance, and a guidance definition can be offered at zero to many task.

c) SPEM 2.0 Profile Notation

Stereotype	Super /metaclass	Keyword	Proprieties	Abstract	Icon
GuidanceDefi nition	Guidance ProcessElement /	GuidanceDefi nition	Profile Category	No	

4.2. The « *ProcessWithMethod* » package

This package describes the using of the development elements and methods describing by a given software process in accordance with the "*ProcessStructure*" concepts. The product of this description is a software process described in a particular context and according to the retained life cycle. Most classes and stereotypes of this package have the suffix "Use". The Use concept means the use of an instance class on a specific case. Thus, we extend SPEM profile with new stereotype "*GuidanceUse*" (**see Figure -7-**).
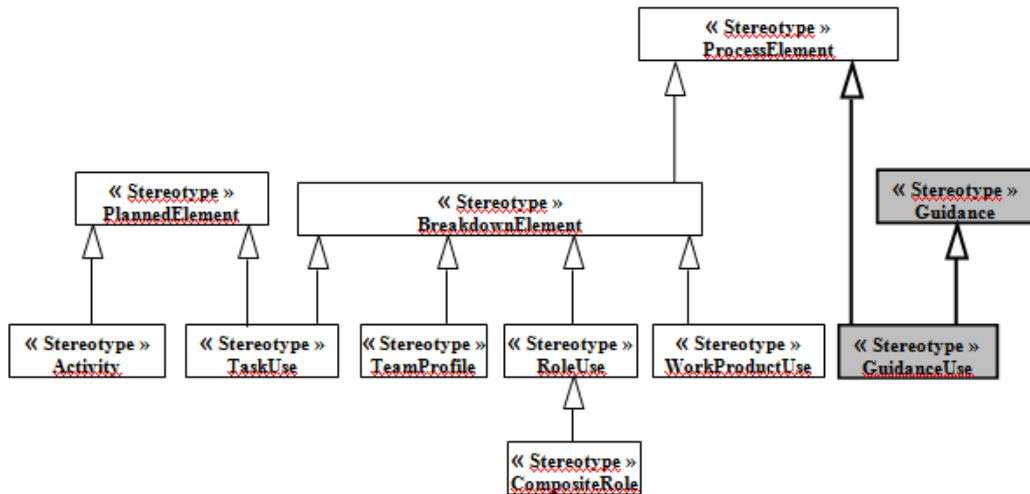

Figure -7- : Extended SPEM 2.0 Profile by the "GuidanceUse" stereotype.

We define the guidance methodologies for particular uses in "ProcessWithMethod" through the class "GuidanceUse". This stereotype role will allow the description of the guidance used in particular contexts. Its structural relation and its proper association sets are described in the following figure (**see Figure -8-**).

To be conform with the structural relation between the MC and PWM, we maintain an association between Use classes and Definition classes as follows
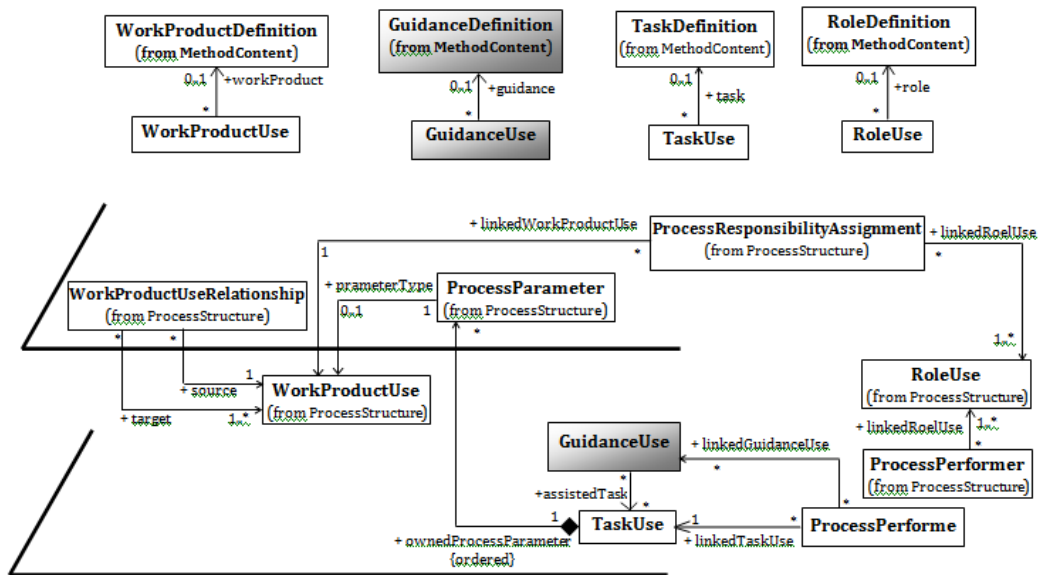
Figure-8- : The main classes and relations of extended package "ProcessWithMethod".

▶ The description of « *GuidanceUse* » *class*.

The "GuidanceUse" semantic and its different associations are as follows.

a) Semantic definition

The «*GuidanceUse»* is a key concept to insure the application of the defined guidance *"GuidanceDefinition"*. It is considered as a reference object for an adapted guidance to a particular context. The use of guidance depends on the description and the evolution of the *"TaskUse"* element as well as the role and the qualifications of the performer applied by "*RoleUse*" element.

b) Associations

Super class: Process Element Guidance.

The «*GuidanceUse*" explicit the adaptive use of the defined guidance in relation to a particular context.
It takes into account the characteristics of a "*TaskUse*" as well as the qualifications and behavior actor "*ProcessPerformer*". This dependency is reflected by the following associations:

- o **Guidance: GuidanceDefinition;** *this* association represents the link between one or more elements *MethodContentUse"* to an element of "*MethodContentDefinition*".

- o **LinkedGuidanceUse: GuidanceUse ;** This association expresses the fact that "*GuidanceUse*" element provides many defined guidance for the developer of the task in relation with its profile. And the developer may have zero or more guidance "*GuidanceUse*" as a development support of its task.

- o **Assisted task: Task Use:** the "*TaskUse*" instance may be assisted by zero to several guidance "*GuidanceUse*", and a "*GuidanceUse*" may be provided at zero or more tasks.

c) SPEM 2.0 Profile Notation

Stereotype	Super /metaclass	Keyword	Proprieties	Abstract	Icon
GuidanceUse	Guidance ProcessElement	Guidance Use	Profile Category	No	

5. CASE STUDY OF THE ADAPTIVE GUIDANCE

The use of our assistance system is illustrated through two concrete situations related to methodological principles and to back return as follow:

To be an example of software procedure related to an activity of development defined through 6 tasks:
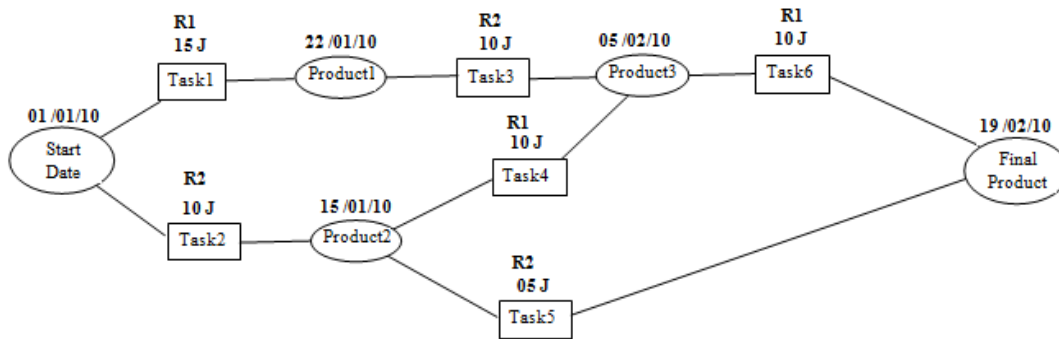

Figure-9- : An Example of Activity Diagram.

<u>*SITUATION 1*</u>**: METHODOLOGICAL ASPECT BY THE RESPECT OF THE ACTION PLAN FOR THE LAUNCH PF THE TASK 6.**

The system of Assistance determines automatically, according to the profile of the developer of retro contextual and adaptive actions to undertake according to a corrective orientation, constructive or specific;

▶ *Corrective Assistance*: the system of assistant analyzes the situation and generates the following message: «the rate of progress (promotion) of task 4 is low(weak) and provokes a non compliance with the term of the end '05/02/2010' ".
The developer and the administrator remain free of the continuation to be given to the error message.

▶ *Constructive Assistance*: the system of assistance analyzes the situation and builds according to the context, one or several possible solutions. For example, For the case of " the situation 1 " with a noticed delay of 05 days:

▶
After analyzing the situation 1, the system makes finding a delay of 05 days on the Task 4. The system then offers the developer a choice of 02 variants of possible solutions.
• *Variant 1*: put back the term of the end of task 4 on 05/02/2010 + 5 days.
• **Variant 2**: strengthen the resources "developer" of task 4 to reduce the remaining duration of 05 days.

The construction is not definitive; it should be validated by developer.

► **Automatic assistance**: the system of assistance analyzes the context, deducts the possible solution and proceeds automatically to the implementation of this one.

In our example: the analysis of the context has identified a low (weak) progress (promotion) rate of task T4, the likely delay, the system of assistance deducts the solution which imposes upon the respect of methodological aspect and to proceed to the automatic update of the action plan by a level-headedness of 05 days of the duration of task 4 " new term shall be 105/02/2010.

SITUATION 2: CASE OF BACK RETURN

In that case, our system of assistance analyzes the plan of evolution, revises the strategy of development and proceeds according to the profile of the developer, to inform the developer or to proceed to an automatic and adaptive intervention for the treatment of the principle of adopted back return.

In our example of software process for landing in a situation of blocking, after 50 % of the progress rate of task 2,
It is to note that it has been stopped launching task 4 and task 5 at the moment when task T2 reaches a 50 % progress.

► **Corrective Assistance**: the system of assistance analyzes the adopted strategy and generates the following message: "the progress rate of task 2 is more than 50 %; the strategy of return back made is inconsistent.

"Reminder: the adopted strategy is to launch task 4 and task 5 at the moment when task T2 reaches a rate of 50 % progress".
The developer and the administrator remain free of the continuation to be given to the error message.

► **Constructive Assistance**: the system of assistance analyzes the adopted strategy and generates the following message: "the progress rate of task 2 is more than 50 %; it would be necessary to cancel the launch of tasks 4 and 5 before relaunching the software process from the last date of launch of task 2.

The construction is not definitive; it should be validated by the developer.

► **Automatic assistance**: the system of assistance analyzes the adopted strategy and proceeds to the cancellation of the current activity " task 2 " and all the activities which depend on it tasks 4 and 5 and proceeds to an automatically and coherent resumption of the development process.

6. CONCLUSION

This paper deals with the problem of adaptive assistance for software process modeling. We have extended the Meta model SPEM with new concepts and stereotypes providing software process modelers with a specific assistance support adapted to the profile of the development context based on the actor, activity and team development models.

This contribution is fulfilled by modeling an adaptive guidance system described by the integration of stereotypes relating to guidance at the packages profile, "*MethodContent*" "*ProcesswWithMethod*", and their description by new stereotyped classes and respective associations.

Our adaptive guidance approach has been implemented through the integration of our concepts to the Eclipse Process Framework EPF / SPEM. So far, there is ongoing progress of improvement.

One Perspective to this work concerns the development of semantic rules allowing to swap through different guidance profiles, either statically by adjustment of guidance parameters or dynamically through the developer behavior.

REFERENCES

[1] B. Combemale, A. Caplain, X. Crégut, B. Coulette.(2006) "Vers une vérification d'un procédé de développement modélisé en SPEM". In: Formalisation des Activités Concurrentes.

[2] Hans-Ulrich Kobialka,(2004) "Supporting the Software Process in A Process-centered Software Engineering environment", Upgrade-cepis.org/issues/2004/5/upgrade-v VOL ; V n° 5.

[3] MALGOUYRES H., MOTET G.(2006) "A UML Consistency Verification Approach Based on Metamodeling Formalization", Symposium on Applied Computing, Dijon, France, ACM publishers.

[4] Pascaline Laure Tchienehom,(2005) "Modèle générique de profils pour la personnalisation de l'accès à l'information". In INFORSID. p.269-284. Grenoble, France.

[5] Hamid Khemissa, (2008) "A Generic assistance system of software process": in International Conference on Software Engineering: Software Engineering. SE 2008.

[6] OMG. Inc.(2008) "Software Process Engineering Meta-Model Specification version 2.0: Formal/2008-04-01".

[7] Cugola G. et al., (1998) "software processes: a Retrospective and a Path to the Future", In software process – Improvement and practice, vol. 4, pp. 101 – 123.

[8] Gruhn V., (2002) "PSSEs, A Brief History and Future Challenges", Annals of S.E. 14(1-4), Kluwer, Academic Publishers, Netherlands, pp 363-382.

[9] Sutton S. & Osterweil L., (1998) "the design of a Next-Generation Process Language". In Proc of the 3rd ACM SIGSOFT / SIGPLAN Symposium. On Software Development Environments.

[10] S. dami et al, (1998) "APEL: a Graphical Yet Executable Formalism for Process Modeling" Kuwler Academic Publisher, pp. 60-96.

[11] Coulette Bernard., Crégut Xavier. et al, (2000) "RHODES, a Process-centered Software Engineering Environment", in Proc. of ICEIS 2000, Stafford, pp 253-260.

[12] Bradshaw J., Editor, (2000) "Handbook of Agent Technology" , MIT Press.

[13] Ledeczi A. et al, (2001) "The Generic Modeling Environment", Workshop on intelligent Signal Processing,.

[14] V. Lestideau, (2002) "Un environnement de déploiement automatique pour les applications à base de composants". ICSSEA 02.

[15] Xavier Thirioux, Benoit Combemale, Xavier Crégut, Pierre-Loïc Garoche. (2007) " A Framework to formalize the MDE Foundations". In : International Workshop on Towers of Models (TOWERS).

[16] Ivan Garcia and Carla Pacheco (2009) "Toward Automated Support for Software Process Improvement Initiatives in Small and Medium Size Enterprises". Book chapter. Software Engineering Research, Management and Applications. Volume 253/2009, pp. 51–58. c_ Springer-Verlag B. ISBN: 978-3-642-05440-2.