



**HAL**  
open science

# PLASTICITY OF A GUIDANCE SYSTEM FOR SOFTWARE PROCESS MODELING

Hamid Khemissa, Mohamed Ahmed-Nacer, Mourad Oussalah

► **To cite this version:**

Hamid Khemissa, Mohamed Ahmed-Nacer, Mourad Oussalah. PLASTICITY OF A GUIDANCE SYSTEM FOR SOFTWARE PROCESS MODELING. The First International Conference on Computer Sciences and Information Technology CoSIT 2014, Sep 2014, Bangalore, India. 10.5121/csit.2014.4905 . hal-01139407

**HAL Id: hal-01139407**

**<https://hal.science/hal-01139407v1>**

Submitted on 29 Apr 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# PLASTICITY OF A GUIDANCE SYSTEM FOR SOFTWARE PROCESS MODELING

Hamid Khemissa<sup>1</sup>, Mohamed Ahmed-nacer<sup>2</sup> and Mourad Oussalah<sup>3</sup>

<sup>1,2</sup> Computer Systems Laboratory, Computer Science Institute,  
USTHB University, Bab Ezzouar Algeria

<sup>1</sup>hkhemissa@hotmail.com,

<sup>2</sup>anacer@cerist.dz

<sup>3</sup> Computer Laboratory Nantes Atlantique, Faculty of science Nantes University  
Mourad.Oussalah@univ-nantes.fr

## ABSTRACT

*The need for adaptive guidance systems is now recognized for all process of software development. The new needs generated by the mobility context for software development led these guidance systems to be adapted for. This paper deals with the plasticity of guidance systems or their ability to be adapted to specific development contexts. We propose a Y description for adaptive guidance. This description focuses on three dimensions defined by the material platform, the adaptation form and provided guidance service. Each dimension considers several factors to deduce automatically the appropriate guidance service to a current development context.*

## KEYWORDS

*Plasticity, Adaptation, Development context, Guidance systems plastic.*

## 1. INTRODUCTION AND PROBLEMATIC

The software development organizations are actually confronted to difficulties regarding the development of their applications. Due to technological progress, the developer is considered nowadays as a mobile actor working in various development context using variable platforms. This trend seems interesting from a user perspective, it poses a new problem in software processes engineering. This concern denotes the adaptation ability to the possible variations of the development context. The objective is to support the process by providing software tools to model, improve, assist and automate development activities [1, 2]. For this purpose, the research in the software processes modeling have known a considerable evolution focusing on defining concepts and objectives for modeling and defining Process-Centered Software Engineering Environments [2,3,4]. They agree on the following goals like to facilitate the comprehension and communication process, to describe clearly the roles, responsibilities and interactions between users, to automate the execution of repetitive tasks that do not require the human actor intervention and finally, whatever the support and the development context used, to provide guidance to actors about modeling and handling a software process. According to the aim and orientation given to the software process, it is possible that other concepts such as strategy, organization and guidance can be described in the software process meta-model.

For this, it is necessary to assist developers and to ensure plasticity of the guidance systems [5, 6] by their ability to adapt to the current development context in respect of their usefulness. Also, usefulness is not limited to performance criteria in the tasks accomplishment, it relates rigorously to satisfaction services offered to developers. By development context, we mean the triplet (material platform, developer profile, activity context). Usefulness refers to the ability of a guidance system to allow the developer to reach his objective preserving consistency and product quality in software development.

In this perspective, a rigorous guidance system targets two basic aspects: 1) The progress control of the software process development regarding the temporal constraints of the activity and the consistency of the results, and 2) the guidance interventions adapted to the specific needs within the development context in progress.

Section 2 of this paper presents a synthesis of similar work and describes the current tend. Section 3 presents our approach of the Y adaptive guidance modeling while section 4 describes the implementation process of the adaptive guidance. Section 5 describes the Plasticity of Guidance Meta model (PGM) and section 6 presents the practical cases study of the adaptive guidance. It ends with a conclusion and future prospects.

## **2. RELATED WORKS AND CURRENT TEND**

Several process-centered environments [7, 8, 9] deal with the assistance aspect in the support of the software product development. However, the provided guidance is not often adapted to the development context profile. The orientations of the guidance are defined on the basis that the human actor, regardless of his profile (qualifications and behavior), has a central role in the progress of the development process.

Among this new generation of the software process engineering, we can invoke the following meta-models and modeling environments: SPEM [10] and APEL [8] considered as the most representative in the software process modeling, RHODES [7][11] that uses basic concepts closest to those introduced by the proposed approach.

SPEM meta-model introduced the concept of "Guidance". According to SPEM, the guidance is a describable element which provides additional information to define the describable elements of modeling. However, the proposed guidance is not suitable to the development context's profile (role, qualifications and behavior). The guidance is rather defined in an intuitive way. ADELE/APEL proposes a global assistance of proscriptive type without considering the development context profile and automates part of the development process using triggers. RHODES/PBOOL+ uses an explicit description of a development process. The activities are associated to a guidance system with various scenarios of possible realization.

An effective support to software process depends on several factors, in particular the personalization and adaptation factor. The definition of a process with an active guidance for automation and coherency control would be effective if it can be adapted to each development context. The platform, tasks and developers characteristics may considerably vary. An improved productivity and development process adaptation would be possible, if a process can be adapted considering the fact that these characteristics can be exploited.

Actually, there are Process Centered Software Engineering Environments (PSEE) allowing changes during the execution, where the developer is in a position to predict the execution model before running it. However, these models do not provide appropriate performance models. Some PSEEs use a guidance description structured in phases like prescribing systems or proactive

systems to control the operations carried out by the developer. Nevertheless, they are essentially limited to the adaptive guidance aspect to current development context.

Taking into account specific criteria for an adaptive guidance, we have classified these limits through several criteria describing explicitly the basic concepts linked to the adaptive guidance [5, 6, 12]. To realize the effectiveness of plasticity concept of the guidance system supported by its adaptation ability to current development context, we refer to the studied meta-models and modeling environments [12, 13]. The selected criteria are defined by:

- ▶ **Global guidance core:** The basic guidance is defined as a global orientations core regardless the profile of both the activity and the actor.
- ▶ **Developer profile oriented guidance:** the guidance orientations are defined on the basis that the human actor, regardless his profile, has a central role in the progress of the development process.
- ▶ **Context development guidance:** The selection of the appropriate type of guidance is more often not adapted nor suitable to a current context.
- ▶ **Guidance types:** 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.
- ▶ **Plasticity of guidance:** the guidance functions are defined and offered on the basis that the human actor always operates in a uniform development context.

To respond to these limits, one currently tries to offer more flexibility in the language of software process modeling. This tendency results in the idea to define interventions of direct and adaptive assistance in particular contexts during the progress of software process. In considering the principal limitations of PSEEs and essential characteristics of our approach in particularly the context adaptation aspect, a comparative table of the studied meta-models is as follows.

Table 1. Comparative table of the studied meta-models.

| <b>Meta model</b><br><b>Criteria</b>       | <b>ADELE/APEL</b>             | <b>RHODES /<br/>PBOOL+</b>              | <b>SPEM</b>                   |
|--|-------------------------------|---|-------------------------------|
| <b>Global guidance core</b>                | Global                        | Global                                  | Global                        |
| <b>Developer profile oriented guidance</b> | Not adapted                   | Considered strategy Model               | Not adapted                   |
| <b>Context development guidance</b>        | Not adapted                   | Adapted                                 | Not adapted                   |
| <b>Guidance types</b>                      | Not invoked                   | Associated with a specific guide system | Intuitive selection           |
| <b>Plasticity of guidance</b>              | Not covered (Single Platform) | Not covered (Single Platform)           | Not covered (Single Platform) |

The current tendency is that developers would like to have integrated environments that are suitable to specific needs according to the characteristics of the development context. However, despite the necessity imposed by technological evolution, the provided efforts to develop such environments remain an insufficient contribution. This generation of guidance environment still interests researchers in defining new concepts and objectives of the software process modeling [4, 14, 15].

Our work proposes an approach to define adaptive guidance modeling in software process. The proposed approach concepts are described through a meta-model denoted PGM (*Plasticity of Guidance Meta model*). The information provided must be adapted to the development context profile. They must guide the developer during the software process development through suitable actions and decisions to undertake with corrective, constructive or automatic intervention [12]. Its adaptation is explicitly described by three plasticity dimensions defined by the development context, the adaptation form and the provided service.

### 3. THE ADAPTIVE GUIDANCE IN Y

A guidance system may be processed in many different ways according to the perspective guidance to provide interveners with development context. Thus, there are generally several possible assistance models, each of them with a particular relevance and need. This vision denotes the plasticity of guidance system, and its ability to adapt to their development context. The plasticity concept describes its capacity to adapt to the intrinsic variations of required conditions in terms of usefulness [16, 17].

In this context, we propose a description in Y of the adaptive guidance. This description will focus on the three considered dimensions. Each dimension considers several factors to deduce automatically the appropriate guidance service according to the current context. It is schematically described as follows:

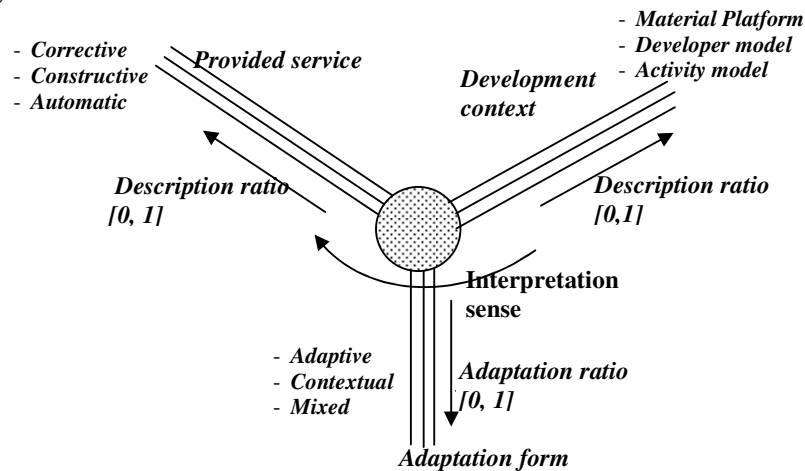


Figure 1. Adaptive guidance description in Y.

The principle of our approach is to generate, from the development context related to the specific data for each defined models according to the retained adaptation form, the guidance interventions (corrective, constructive, automatic) adapted to the current development context.

#### 3.1. The basic conceptual model description

The conceptual model highlights the guidance plasticity aspect through the adaptation form described by the inherent relationship between the three considered dimensions.

##### 3.1.1. The development context

A guidance intervention is provided according to an object or set of objects. An object is associated to:

- ▶ The hardware and software platform: described by the computing resources, the software services and the interaction and communication modes.
- ▶ The activity model: models the structure and the workflow, they are defined by a progression mode in the activity ensuring that all tasks can be performed under control in a preset order established by the designer and a temporal progression mode specifying deadlines for completion.
- ▶ The developer model: defines the specific properties of each developer. These properties can be either static or dynamic. The static aspect refers to the user characteristic as his role, his business competence and his familiarity with the software process. The dynamic aspect refers to the behavior of using the guidance system, by the fact to execute, to define or to complete the software process resource and the user's reaction to a guidance message.

The description performance rate of these factors is evaluated by considering each identified object as concepts, principles, procedures, and resources. These guidance objects represent the basis of different guidance interventions related to a particular situation. This performance serves as the selection of the adaptation form to retain and guidance service to provide to the user.

### 3.1.2. The adaptation form

Each guidance intervention is done according to the retained adaptation form. It relates to a specific situation described by the development context description. Our modeling approach allows the following guidance adaptation forms:

- ▶ **Contextual guidance:** intervention is provided dynamically according to the material platform and activity models and the state of the process. The adaptation rate is related to the model description rate of the activity and the material platform. The guidance intervention doesn't consider the developer model (e.g.: to avoid inconsistency during the affectation of a resource).
- ▶ **Adaptive guidance:** intervention is provided according to the developer model and the material platform specificity (e.g.: the user asks for explanations on his choice). The adaptation rate is related to the developer and material platform models description.
- ▶ **Mixed guidance:** intervention is provided according to the development context (e.g.: to guide the developer on the sequencing principle during the software process progression). This form describes the highest adaptation rate. This rate is evaluated on the basis of the developer, activity and material platform models description.

The adaptation form performance is described by a strong coupling between the development environment and guidance system. It determines the relevance and precision of the guidance provided to developers.

This criterion is directly related with the adaptive guidance system concept. Through a strong coupling, the system would deduce the guidance context and can therefore extract useful and helpful information to the user.

### 3.1.3. The provided service

The guidance system offers several service types in relation to a defined context by the current development and adaptive form. The provided services are corrective, constructive or automatic order.

- ▶ **Control and taking corrective initiative:** protect the user of his own initiatives when they are inadequate under progress.
- ▶ **Control and taking constructive initiative:** the ability to take positive initiatives, executing and combining the performance of operations without the user intervention.

The guidance adaptation performance associated to a development environment is done by enrichment or reduction of the possible offers of the guidance. Among these offers, we have:

- **The directive guidance:** to show the developer how to execute a task by an adaptive control of the guidance system, specifying the steps of an activity or the whole process development.
- **Retroaction,** to offer the developer more information on the activities context (e.g.: new available resources) or on the progress state of his work (progression of an activity).
- **Explanation,** to offer explanations about a guidance object at the request developer. (e.g.: the activities coordination of the software process).
- **Reminding,** to remind the developer some principles or procedures on the sequencing of the activities or their activation conditions when the system detects a conflict or inconsistency.
- ▶ **Automatic guidance:** analyze the impact projection to define the solution to consider in order to avoid deadlocks or delays, by the fact to start, suspend, discontinue or continue ongoing actions to avoid conflict.

These services can be combined. They may be temporary, permanent or left under the developer control.

The usefulness rate is evaluated by the degree of the performance description of the development context and adaptation form.

## 4. THE ADAPTIVE GUIDANCE FUNCTIONING

The implementation of the proposed adaptive guidance is done according to the interpretation sense (see Figure 1). The selection of the adaptation form is relatively based on the description rate of the development context elements. The provided service will focus on the retained adaptation form, relatively to the concepts' interpretation related to the development context. The adaptation of guidance system to the development context can only concern a subset of the latter. In this case, we will talk about a reduced service associated to the provided guidance linked to the current context.

In all cases, the operating strategy in the adaptation is done by the reduction or enrichment of the provided guidance service. Intuitively, we consider the three following adaptation strategies:

- ▶ **Service plus (or enrichment):** is a strategy to enhance the offered guidance services to support an expressed description of the development context.

- ▶ Service minus (or reduction): is to remove guidance items due to a limited or non-critical description of the context.
- ▶ Service poly: generates several possible forms of service. This strategy is supported by the performance rate principle of the provided services.

These Strategies are accomplished according to the political autonomy given to the guidance system in respect of the context conditions and the developer choice. The choice of this policy is made with regard to the performance criteria of the three guidance dimensions. The performance degree of each component varies from 0 to 1. The implementation of this policy is based on an adaptation mode expressed by a set of rules of the ECA form (Event, Conditions, Actions). For each Event, if required Conditions then propose Actions.

#### 4.1. The instantiation process

The instantiation process of the proposed generic model will focus on concrete specific use cases. It described how to generate the guidance service in adequacy with the current development context through concrete situations.

**Situation 1:** to support the progression of high performance developer which evolves on an average order platform and takes in charge a simple activity. Therefore, these two factors are practically without effect, the adaptation of the provided service, based on the developer performance choice, reaches an adaptive guidance of a corrective order (see Figure 2).

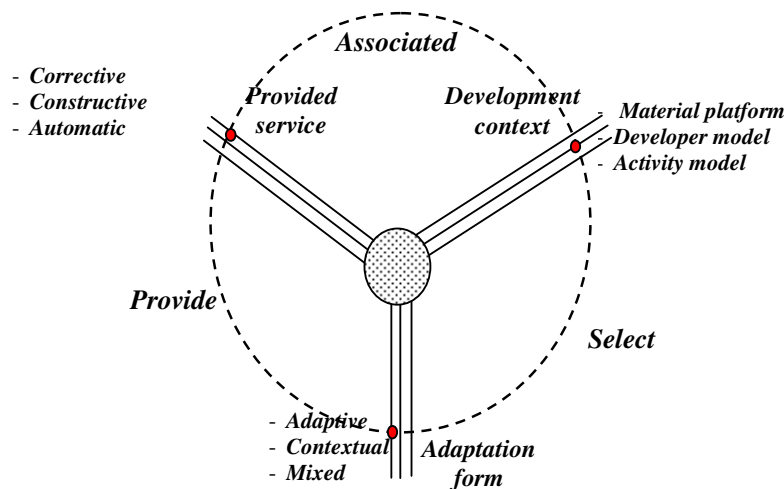


Figure 2. Description of the adaptation scenario : situation 1.

**Situation 2:** Assuming that the development context migrates toward a development platform relatively limited and that the two other factors are always of an average order and practically without effect. The adaptation of the provided service will rather be on a contextual form associated to corrective as well as constructive guidance (see Figure 3).



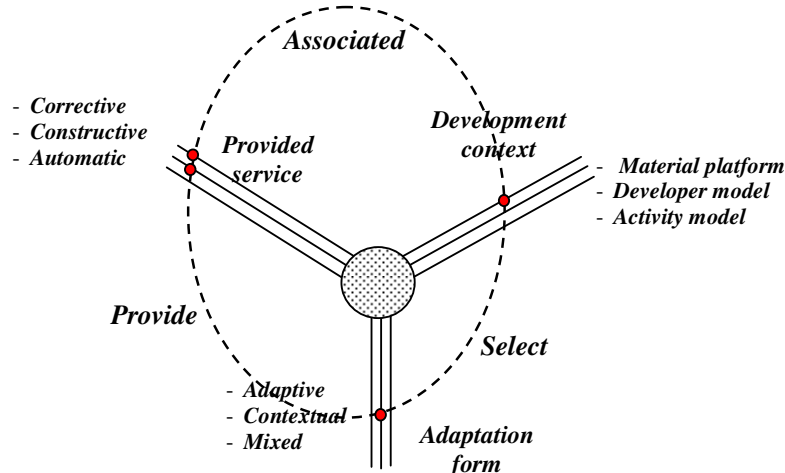


Figure 3. Description of the adaptation scenario : situation 2.

## 5. PLASTICITY OF GUIDANCE META MODEL

Our modeling approach PGM (Plasticity of Guidance Meta model) is defined with reference to the identified limitations of studied PSEEs. The essential characteristic of our approach is to consider the plasticity principle in the development context, defined by the description of its three models [15, 17].

In this context, our meta-model is based on the conceptual model of a software process enriched by the plasticity of the adaptive guidance element. It controls the smooth running of the activities and provides adaptive guidance to the development context.

The Adaptive guidance management addresses the three defined dimensions by the development context, adaptation form and provided guidance service. Each dimension considers several factors to deduce automatically the appropriate guidance service to the current context. It has an operating strategy supported by three services.

The first service 'Service Plus' role is to enhance the guidance function to support the current situation. The second service 'Service Minus' is to adapt by reducing the guidance function to a particular context. Finally, the third service 'Service Poly' generates, according to the current context, the most suitable form of the offered guidance function.

The guidance strategy evolves according to the political autonomy given to the guidance system respecting the application conditions. The implementation of this policy is based on an adaptation mode expressed by a set of rules of ECA form (Event, Conditions, Actions). For each in the execution context, if required conditions related to the context and the adaptation form then launch guidance strategy to generate the most appropriate service.

The proposed meta-model aims to generate the adapted guidance interventions to the development context in relation to the considered properties and specific data for each defined model (see Figure 4).

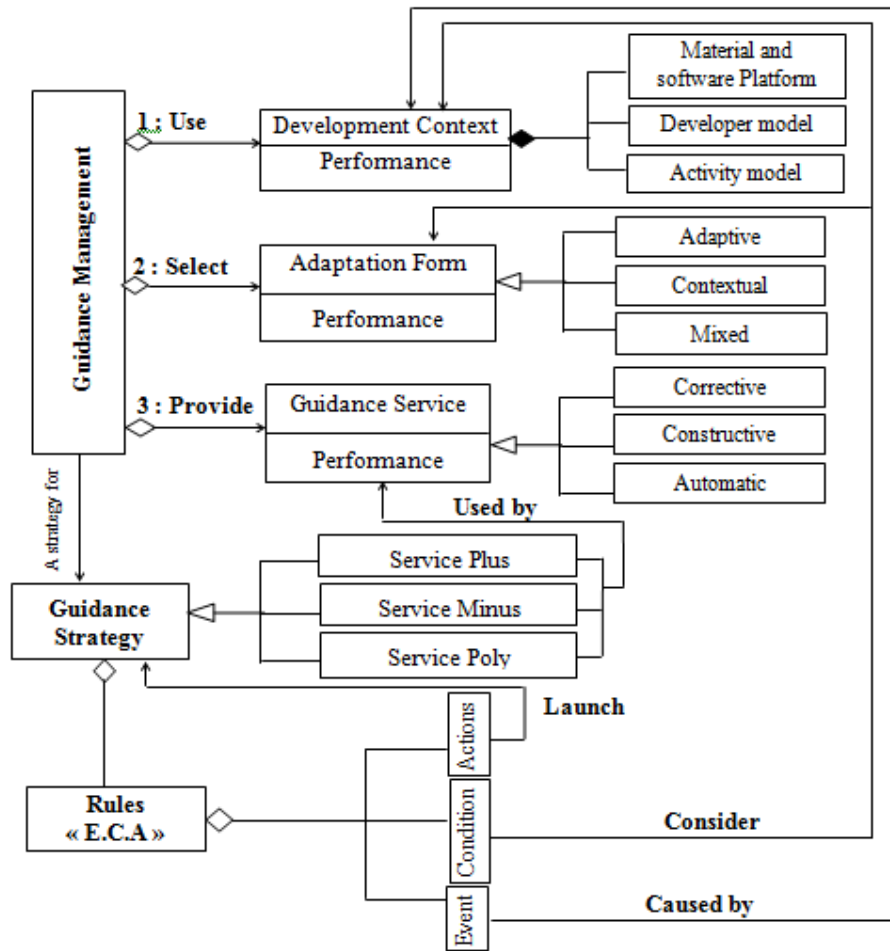


Figure 4. Plasticity of Guidance Meta model.

## 6. THE PRACTICAL INTERPRETATION

Considering the software process model "Activity test", the process "Activity test" in the software development is composed of several types of tests such as: Integration test and Unitary test. Each receives as input a test plan and provides a test report. For each type of test, there is a manager, responsible of the execution.

The activity process "Activity test" is described by a performing tree given in Figure 5. We notice that the activity test starts the execution of subactivities "Unitary test" then "Integration test". The unitary test launches in parallel the execution of tasks "Test unit".

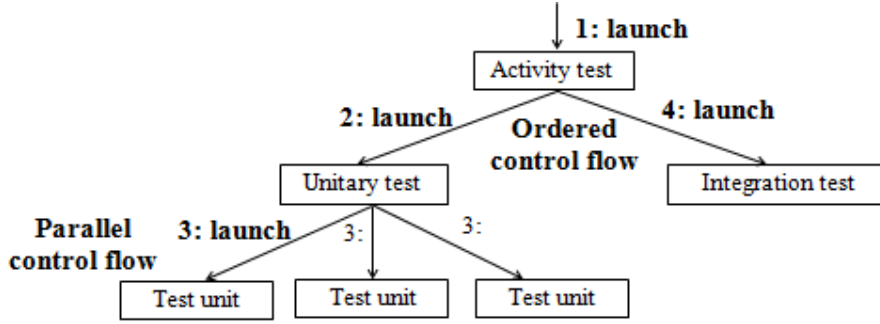


Figure 5. The activity test process.

To simplify our example, we consider the execution process of the unitary component test. The application of the activity “Unitary test”, requires the list of components. It calls the tool that will create the necessary environment to carry out the actual execution of the “Unitary test”, as the state diagram, the test variables, etc. ... the activity "Unitary test" launches in parallel the different tasks "Test unit" where an event signals the beginning of the “Test unit” execution. Finally, the ended event is broadcast.

The adaptive execution process of the activity "Unitary test", regarding our adaptive guidance approach is described by Figure 6.

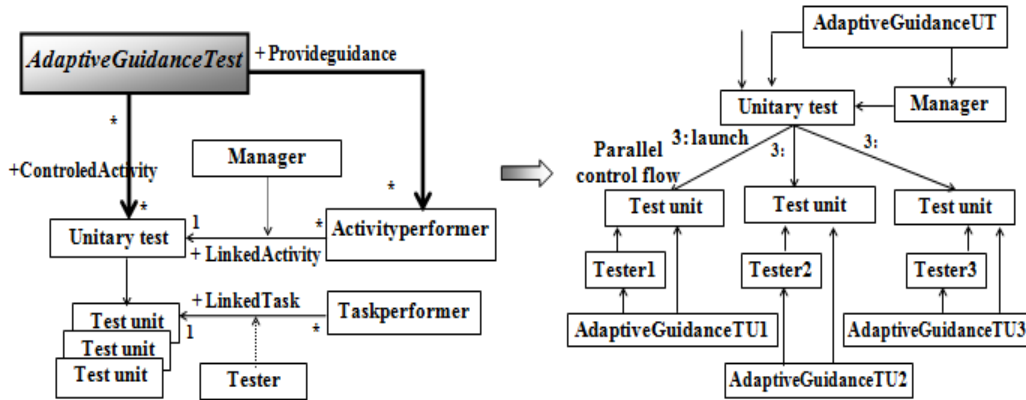


Figure 6. The adaptive execution process “ Unitary test”.

The adaptive guidance is linked to the manager or to each tester according to the current development context profile defined by its material platform, the activity model and developer model. We explain this adaptive approach through the following situation; the testers have the same role “test unit” with identical activity model. However, the developer’s qualification and the material platform specificity differ from one development context to another. According to the current context, it happens to enrich or reduce the appropriate guidance intervention or generate several possible forms of guidance services.

We consider three situations with tester’s qualification defined respectively as high, medium, and low. The study case is related to launch the test unit without having all the input data, by selecting the appropriate test variables and generating the unit test report. The adaptive guidance process related to each qualification case is described as follows:

1. For a development context with high qualification tester and a high material platform performance: the tester starts the test unit process on the basis of the defined plan by taking his proper initiatives. The development context evaluation allows deducing the adaptation form to retain and the guidance service to provide. In this case, we adopt the adaptation guidance form and the provided guidance intervention is thus of a corrective order. The corrective intervention is provided to inform the manager of the setback and remind him of the corresponding unitary test diagram. The manager remains free to take into account the intervention.
2. For a development context with an average skill tester and an acceptable material platform performance: the tester starts the test unit process by applying rigorously the defined test plan. The evaluation of such context results in a contextual guidance form and the provided guidance intervention is thus of a constructive order. The guidance system analyzes the current context of the task, evaluates the impact and consequence of the delay caused in comparison with possible margins and offers a possible solution to the manager (solution: the guidance proposes to cancel the launch of the current test unit and generates a new execution plan according to the rate of delay and possible margins). The construction solution is not definite; it should be validated by the manager.
3. For a development context with a low qualification tester and a reduced material platform performance: the tester starts the test unit process by applying reliably the defined test plan. The development context evaluation results in a mixed guidance adaption form and the provided guidance intervention is thus of an automatic order. The guidance system analyzes the current context, cancels the launch of the "test unit" task, evaluates the impact and consequence of the delay caused in comparison with the possible margins and automatically updates the execution plan of "unitary test" activity.

## 6.1. The digital application

The practical definition of the adaptive guidance type for each considered profile is deduced by a quantitative process of the characteristics in relation to the basic models (materiel platform, activity model, developer model). The considered example is processed as follows.

Each profile is semantically described in table (see Table 2). The semantics evaluation and the weighting are determined by the project manager under the specification of an ongoing project [14]. To scan the semantics evaluation, we associate the weighting related to the interest granted to each attribute.

Table 2. The profiles evaluation.

| Development context      | Features                      | Context Profile 1 | Context Profile 2 | Context Profile 3 | Context Profile 4 | Context Profile 5 | W[i]      |
|--------------------------|-------------------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-----------|
| <b>Material Platform</b> | Development System Constraint | Low               | Medium            | High              | High              | Low               | <b>P2</b> |
|                          | Software Tools                | Low               | Medium            | Low               | Provided          | Provided          | <b>P1</b> |
|                          | Memory Constraint             | High              | Medium            | Medium            | High              | Medium            | <b>P3</b> |
| <b>Developer</b>         | Role                          | No effect         | Classic           | Critique          | Critique          | No effect         | <b>P4</b> |
|                          | Competence                    | High              | Medium            | Low               | Low               | High              | <b>P1</b> |

|                       |                                   |                  |            |            |            |            |           |
|-----------------------|-----------------------------------|------------------|------------|------------|------------|------------|-----------|
| <b>Model</b>          | Familiarity with Software Process | Quite Acceptable | Medium     | Low        | Low        | Acceptable | <b>P1</b> |
|                       | Behavior for assistance           | Most Appropriate | Satisfying | Inadequate | Adequate   | Inadequate | <b>P2</b> |
| <b>Activity Model</b> | Density of tasks in the activity  | Acceptable       | Acceptable | Acceptable | Acceptable | Acceptable | <b>P3</b> |
|                       | Complexity Level                  | Medium           | Medium     | Medium     | Medium     | Medium     | <b>P2</b> |

With  $W[i] \in [1, 5]$ . Where  $P_i$  represents the computing value.

Considering the similar principle such as the COCOMO model, the quantification of each profile's feature is on the data range] 0, 2 [, (see Table 3). This quantification is usually based on the impact of each feature.

It is usually done through three levels, described by high, medium or low contribution, applying the following rules:

1: middle order impact / <1: positive impact / >1: negative impact.

Table 3. The profiles quantification.

| Development context      | Features                          | Context Profile 1 | Context Profile 2 | Context Profile 3 | Context Profile 4 | Context Profile 5 | W[i]     |
|--------------------------|-----------------------------------|-------------------|-------------------|-------------------|-------------------|-------------------|----------|
| <b>Material Platform</b> | Development System Constraint     | <b>0.75</b>       | <b>1.00</b>       | <b>1.30</b>       | <b>1.30</b>       | <b>0.75</b>       | <b>1</b> |
|                          | Software Tools                    | <b>1.20</b>       | <b>1.00</b>       | <b>1.25</b>       | <b>0.80</b>       | <b>0.80</b>       | <b>3</b> |
|                          | Memory Constraint                 | <b>1.40</b>       | <b>1.00</b>       | <b>1.00</b>       | <b>1.60</b>       | <b>1.00</b>       | <b>2</b> |
| <b>Developer Model</b>   | Role                              | <b>0.40</b>       | <b>1.00</b>       | <b>1.90</b>       | <b>1.70</b>       | <b>0.40</b>       | <b>4</b> |
|                          | Competence                        | <b>0.20</b>       | <b>1.00</b>       | <b>1.70</b>       | <b>1.70</b>       | <b>0.25</b>       | <b>3</b> |
|                          | Familiarity with Software Process | <b>0.40</b>       | <b>1.00</b>       | <b>1.60</b>       | <b>1.60</b>       | <b>0.30</b>       | <b>3</b> |
|                          | Behavior for assistance           | <b>0.20</b>       | <b>0.80</b>       | <b>1.70</b>       | <b>0.75</b>       | <b>1.60</b>       | <b>1</b> |
| <b>Activity Model</b>    | Density of tasks in the activity  | <b>0.80</b>       | <b>0.80</b>       | <b>0.80</b>       | <b>0.80</b>       | <b>0.80</b>       | <b>2</b> |
|                          | Complexity Level                  | <b>1.00</b>       | <b>1.00</b>       | <b>1.00</b>       | <b>1.00</b>       | <b>1.00</b>       | <b>1</b> |

In this stage of profiles' process, and in case of simple profiles' samples, we can proceed to associate each considered development context profile to the appropriate guidance adaptation form and guidance service.

The guidance profile (GP) associated to each profile class is based on the following formula:

$$GP(P_x) = \frac{\sum A_i W_i}{2 * \sum W_i} \text{ avec } i=1 \text{ to } n$$

With :

$A_i$ : the feature value.

$W_i$ : the associated weighting.

$P_x$  : the associated profile.

The adaptation form and the guidance profile of each considered development context profile based on the evaluation of each model and GP value is given by (see Table 4).

Table 4. The associate guidance profile.

|                                    | <b>Context Profile 1</b> | <b>Context Profile 2</b> | <b>Context Profile 3</b> | <b>Context Profile 4</b> | <b>Context Profile5</b> |
|------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|-------------------------|
| <b>Associated Adaptation form</b>  | Adaptive                 | Contextual               | Mixed                    | Mixed                    | Adaptive                |
| <b>Guidance Profile (GP)</b>       | 0.333                    | 0.485                    | 0.721                    | 0.673                    | 0.315                   |
| <b>Associated guidance profile</b> | Corrective               | Constructive             | Automatic                | Automatic                | Corrective              |

It should be noted that the value of GP ranged from 0 to 1 and the range associated with each type of guidance is defined by the fixed limits to each guidance type. If the range of corrective guidance is fixed between 0 and 0.35 and the range of the constructive guidance is between 0.36 and 0.65, we automatically associate a corrective guidance to profile P1 and P5, and a constructive guidance to profile P2 and automatic guidance to profile P3 and P4.

However, in case of a very important population, and for the aim of optimizing profile classes, it is recommended to proceed in the gathering and classification of the provided development profile and reasoning in relation to generated classes.

## 7. CONCLUSION

Our main purpose in this article is to propose a plasticity of a guidance system for software process modeling. This plasticity is highlighted through a description in Y of our adaptive guidance. This description will focus on three dimensions defined by the material platform, the adaptation form and the provided service. Each dimension considers several factors to deduce automatically the appropriate guidance service according to the current context. The proposed approach concepts are described through a meta-model denoted PGM (*Plasticity of Guidance Meta model*). The proposed meta-model aims to generate the adapted guidance interventions to the development context in relation to the considered properties and specific data for each defined model.

The operating strategy in the adaptation is done by the reduction or enrichment of the provided guidance service. Intuitively, we consider the three adaptation strategies (Service Plus, Service Minus, and Service Poly). The guidance strategy evolves according to the political autonomy given to the guidance system respecting the application conditions. The implementation of this policy is based on an adaptation mode expressed by a set of rules of ECA form.

A perspective to this work concerns, at first, the necessity to estimate the productivity and cost due to the adaptation of guidance system.

In a second step, we will ensure the development of semantic rules which allow swapping through different guidance profiles, either statically by adjustment of guidance parameters or dynamically through the performer behavior.

## REFERENCES

- [1] Ivan Garcia and Carla Pacheco « Toward Automated Support for Software Process Improvement Initiatives in Small and Medium Size Enterprises ». Book chapter. Software Engineering Research, Management and Applications 2009 Volume 253/2009, pp. 51–58. c\_ Springer-Verlag Berlin Heidelberg 2009. ISBN: 978-3-642-05440-2.
- [2] Kirk, D.C, MacDonell, S.G., & Tempero, E. 2009 Modeling software processes - a focus on objectives, in Proceedings of the Onward, 2009. Conference. Orlando FL, USA, ACM Press, pp.941-948.
- [3] Benoît COMBEMALE, Xavier CRÉGUT, Alain CAPLAIN et Bernard COULETTE. Towards a rigorous process modeling with spem. Dans ICEIS (3), pages 530–533, 2006
- [4] Hamid Khemissa, Mohamed Ahmed-Nacer, Mourad Daoudi, 2008. A Generic assistance system of software process. In International Conference on Software Engineering: Software Engineering, SE 2008, Feb 12-14-2008, Innsbruck, Austria.
- [5] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Souchon, N., Bouillon, L., Florins, M., Vanderdonck, J.: Plasticity of User Interfaces: A Revised Reference Framework. In: TAMODIA 2002 (2002).
- [6] Joëlle Coutaz, EICS '10. User interface plasticity: model driven engineering to the limit!. Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems. June 2010.
- [7] Coulette B., Crégut X., Dong T. B. T. and Tran D. T., “RHODES, a Process Component Centered Software Engineering Environment”, ICEIS2000, 2nd International Conference on Enterprise Information Systems, Stafford, pp 253-260, July 2000.
- [8] Jacky Estublier, Jorge Villalobos, Tuyet Le Anh, Sonia Jamal-Sanlaville and German Vega. An Approach and Framework for Extensible Process Support System. In Proceedings 9th European Workshop on Software Process Technology (EWSPT 2003), Helsinki, Finland, 2003-09-01.
- [9] Hans-Ulrich Kobialka, « Supporting the Software Process in A Process-centered Software Engineering environment », Upgrade-cepis.org/issues/2004/5/upgrade-v VOL; V n° 5 October 2004.
- [10] OMG. Inc. Software and System Process Engineering Meta-Model Specification version 2.0: Formal/2008-04-01.
- [11] Tran Hanh Nhi, Bernard Coulette, Xavier Crégut, Thuy Dong Thi Bich, Thu Tran Dan. Modélisation du méta-procédé RHODES avec SPEM. Dans : Recherche Informatique Vietnam-Francophone (RIVF'03), Hanoi, Vietnam, 2003.
- [12] Hamid khemissa, Mohamed ahmed nacer & Mourad Oussalah «Adaptive Guidance System for SPEM ». The First International Conférence on Information Technology Convergence and Services; ITCS, SIP, JSE 2012 pp. 429-441, Bangalore, India.
- [13] Hamid Khemissa, Mohamed Ahmed-Nacer, Mourad Oussalah «Adaptive Guidance based on Context Profile for Software Process Modeling». Information Technology and Computer Science, , July 2012 in MECS 2012.
- [14] Grambow, Gregor and Oberhauser, Roy and Reichert, Manfred (2011) Enabling Automatic Process-aware Collaboration Support in Software Engineering Projects. In: Selected Papers of the ICISOFT'11 Conference. Communications in Computer and Information Science(CCIS).
- [15] Clarke, Paul and O'Connor, Rory (2011) An approach to evaluating software process adaptation. In: 11th International SPICE Conference on Process Improvement and Capability dEtermination, 30 May - 1 jun 2011, Dublin, Ireland. ISBN 978-3-642-21233-8.
- [16] Sottet, J.-S., Calvary, G., Coutaz, J., Favre, J.-M. A Model-Driven Engineering Approach for the Usability of User Interfaces. In Proc. Engineering Interactive Systems (EIS2007), J. Gulliksen et al. (eds), LNCS 4940, (2007), 140-157
- [17] Ferry, N. Hourdin, G., Lavirotte, S., Rey, G., Tigli, J.- Y., Riveill, M. Models at Runtime: Service for Device Composition and Adaptation. In 4th International Workshop Models@run.time, Models 2009(MRT09).

**AUTHORS**

**Hamid Khemissa** is a full associate professor at Computer Systems Department, Faculty of Electronics and Computer Science, USTHB University, Algiers. He is member of the software engineering team at computer system laboratory LSI, USTHB. His current research interests include Software Process Modeling and Software Modeling Assistance.

**Mourad Chabane Oussalah** is a full Professor of Computer Science at the University of Nantes and the chief of the software architecture modeling Team. His research concerns software architecture, object architecture and their evolution. He worked on several European projects (Esprit, Ist, ...). He is (and was) the leader of national project (France Telecom, Bouygues telecom, Aker-Yard-STX, ...). He earned a BS degree in Mathematics in 1983, and Habilitation thesis from the University of Montpellier in 1992.

**Mohamed Ahmes-Nacer** is a full Professor at USTHB (Algiers's University). He is in charge of the software engineering team. He published extensively and his current research interests include process modeling, information systems, software architecture based components and service web development.