ADAPTIVE CONFIGURATION META-MODEL OF A GUIDANCE PROCESS

Hamid Khemissa[1] and Mourad Oussalah[2]

[1]Computer Systems Laboratory, Computer Science Institute, USTHB University, Bab Ezzouar Algeria
[2]Computer Laboratory Nantes Atlantique, Faculty of science Nantes University

ABSTRACT

The current technology tend leads us to recognize the need for adaptive guidance process for all process of software development. The new needs generated by the mobility context for software development led these guidance processes to be adapted. In order to improve the performance of the deployed software development, it is useful to manage the configuration of its evolving aspects. This paper deals with the configuration management of guidance process or its ability to be adapted to specific development contexts. The proposed adaptive configuration Meta-model is worked out on the basis of a Y description for adaptive guidance process. This description focuses on three dimensions defined by the material/software platform, the adaptation form and provided guidance service. Each dimension considers several factors to develop a coherent configuration strategy and provide automatically the appropriate guidance process to a current development context.

KEYWORDS

Guidance process, Configuration process, Adaptive meta-model, Development context.

1. INTRODUCTION AND CURRENT TEND

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's 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 software process by providing software tools to model, improve, assist and automate development activities [1, 2]. 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 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 [2,3,4].

For this, it is necessary to assist developers and to ensure configuration management of the guidance processes [5, 6] by their ability to adapt to the current development context in respect of their usefulness. By development context, we mean the triplet (material/software platform, developer profile, activity context). Usefulness refers to the ability of a guidance process to allow the developer to reach his objective preserving consistency and product quality in software development.

DOI:10.5121/ijcsit.2016.8312

In this perspective, a rigorous guidance process 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.

Several process-centered environments [7, 8, 9] deal with the guidance aspect in the support of the software product development. Some Process Centered Software Engineering Environments (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. 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 behaviour), has a central role in the progress of the development process.

Among this new generation of 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 in 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 behaviour). The guidance is rather defined in an intuitive way. ADELE/APEL proposes a global guidance 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, task and developer factors may considerably vary.

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 the configuration concept of the guidance process supported by its adaptation ability to current development context, we refer to the studied meta-models and modeling environments [12, 13, 14, 15, 16]. The selected criteria are defined by:

- Global guidance core: The basic guidance is defined as a global orientation 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 intuitive way. It depends on the experience and on the informal personality of the project manager.
- Configuration of guidance process: 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 adaptive guidance 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.

Meta model **Criteria**	**ADELE/APEL**	**RHODES / PBOOL+**	**SPEM**
Global guidance core	Global	Global	Global
Developer profile oriented guidance	Not adapted	Considered strategy Model	Not adapted
Context development guidance	Not adapted	Adapted	Not adapted
Guidance types	Not invoked	Associated with a specific guide system	Intuitive selection
Configuration of guidance process	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 development context factors. 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, 17, 18]. Our work proposes an approach to define configuration of an adaptive guidance modeling in software process. The concepts of proposed approach are described through a meta-model denoted CGPM (*Configuration of Guidance Process 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]. The three dimensions defined by the development context, the adaptation form and the provided service explicitly describe our guidance process adaptation.

Section 2 of this paper presents our approach of the Y adaptive guidance modeling. Section 3 describes the configuration management usefulness and the generic view of its operational model, while section 4 describes the configuration management of the guidance process, as well as the Configuration of Guidance Process Meta-model (CGPM) and section 5 presents the practical cases study of the adaptive guidance process. It ends with a conclusion and future prospects.

2. THE ADAPTIVE GUIDANCE IN Y

A guidance process may be processed in many different ways according to the perspective guidance to provide interveners with development context. Thus, there are generally several possible guidance models, each of them with a particular relevance and need. This vision denotes the configuration of guidance process, and its ability to adapt to its development context.

The configuration concept describes its capacity to adapt to the intrinsic variations of required conditions in terms of usefulness [16, 19, 20].

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. Schematically, we describe it as follow

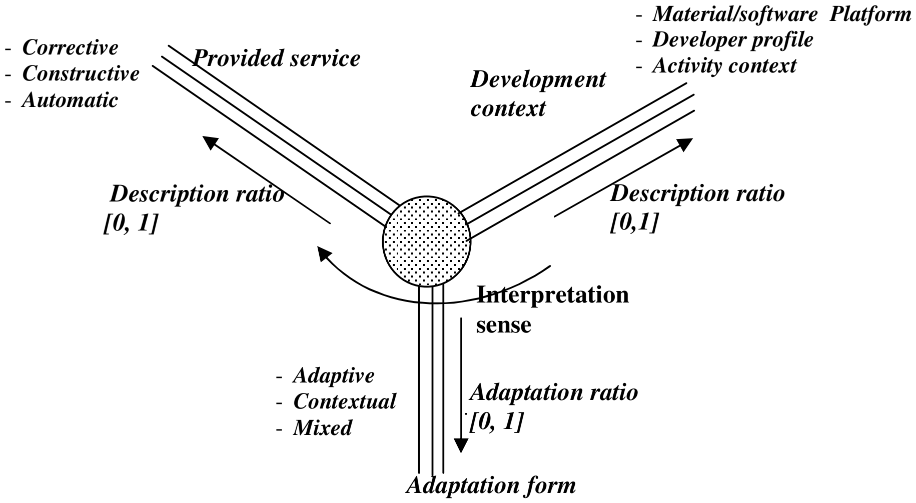

Figure 1. Adaptive guidance description in Y.

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.

2.1. The basic conceptual model description

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

1. The Development Context

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

- The material/software platform: described by the computing resources, the software services as well as the interaction and communication modes.
- The activity context: 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 profile: 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 behaviour of using the guidance service, 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.

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/software platform and activity models. The adaptation rate is related to the model description rate of the activity and the material/software platform. The guidance intervention does not 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/software 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/software platform models description.

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

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

3. THE PROVIDED SERVICE

The guidance process offers several service types in relation to a defined context by the current development and adaptive form. The provided guidance 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 guidance 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.

3. WHY PROCESS CONFIGURATION MANAGEMENT

The configuration management is a constant concern of the software development community, classified as a fundamental discipline to software process development, and necessary for providing all the time a coherent system support regardless the changes of the different development context. This management process seeks adapted tools which allow to supervise any entity evolution under the pre-established norms and rules of a software process. The tools and techniques of initiating, evaluating and controlling change in the software development environment where any entity managed in the software engineering process can potentially be considered under configuration management control. The main purpose of a process management configuration targets the establishment of the coherent support process synchronizing with possible evolutions while maintaining the quality continuity of software process development [14, 21].

The configuration management is one of 17 process of ISO 12207 (Software life cycle support processes) [22]. A specific standard, ISO 10007 is devoted to this process. This management allows to identify, manage and control configuration items throughout the software development process cycle. [23]. However, all the existing standards are still powerless to promote the best practices in the development area.

The dynamic aspect of the configuration management process results from a development perspective of a successive static description in time. It enables to coordinate the development activities and supervise each profile entity evolution over the elaboration of software process to promote coherence with the provided support. In an effective dynamic environment, efficient and flexible guidance process is a suitable support for modeling and handling software process. The guidance process management is an overall task supporting the whole lifecycle of software process. Therefore, a guidance process needs to be planned, adapted and controlled in terms of user's requirements. The operational process of the guidance is analyzed to identify each time the guidance needs in order to improve the performance of the deployed software process. This performance is due to innovations produced by developers, who are expecting to benefit from using support services. This means that we should develop the configuration that fulfills their exact needs, leading them to have a trend to benefit from a support solution, to preserve consistency and product quality in software development [15, 16].

The configuration management plan describes the application scope and key terms. It identifies the responsible who achieves the planned configuration management activities (Who manages?), identifies the management activities to be performed (What activities?). Then identifies tools and techniques required for the configuration planning (HOW planned?), establishes the sequence and coordination of the management configuration activities (When scheduled?). Finally, identifies activities and responsibilities on how the SCMP will be held active during the development lifecycle. As regard the implementation of its process, the required information should be collected by identifying the configuration items to be applied to the configuration management and its application constraints. As well as, to identify the need for a change and how to provide a service support. Moreover, assumptions that might have an impact on schedule and ability to perform defined configuration activities [24].

The operational model of a configuration process is based on the fact that the software configuration management plan might be extended or revised during the rest of the project following any change at the developers' request or by a trigger event that is assessed against the project goals.

This configuration process is an application developed using a configuration server and based on a set of constraints and / or preferences described by the characteristics of the current development context. If one of these constraints and / or preferences causes a change or an evolution notified event then it might mean that we do evolve the current configuration either statically (inactive process) or dynamically (without interrupting the process progress). The purpose of this activity is proceeding, every time, adaptation of the process support to the various changes in the development context in order to ensure its deployment to software process. The functional model description of the configuration method is given in the following figure:

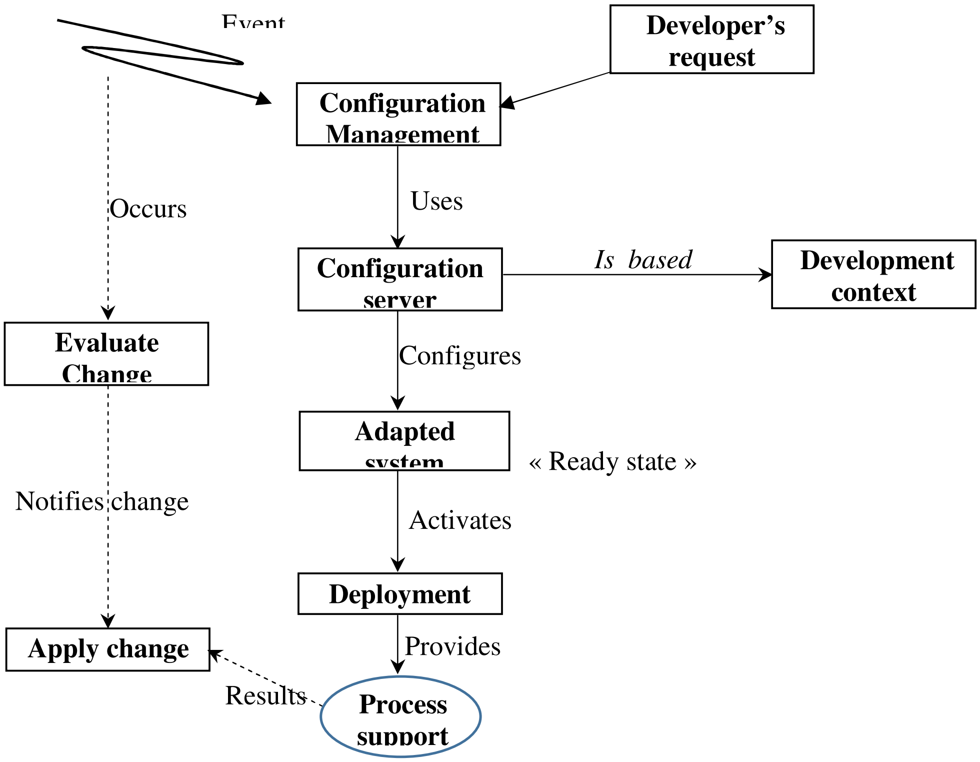

Figure 2. The operational model of the configuration process.

4. THE CONFIGURATION MANAGEMENT OF A GUIDANCE PROCESS

The management discipline within the software engineering process allows to develop a reference basis to any planned configuration. The configuration management is a complex activity that covers all the steps in a software process, from the conceptual aspects to its configuration and installation. It includes the initiation techniques and changes control that manage the software process progress [14, 15, 16].

The role of configuration management is to identify the configuration items, manage the sequence and ensure the evolution and any likely change. These activities are usually treated in different ways depending on the nature and specificity of the objective [15, 16, 25, 26].
In this context, the configuration management of a software process and particularly a guidance process is a configuration mechanism rigorously linked to the evolution of all development context factors. This evolution process is influenced by the temporal and contextual elements. Therefore, it is necessary to define the participating elements and trigger the configuration of a guidance process, and clarify their functional interdependencies.

We describe graphically the configuration basis of our guidance process defined through five interrelated elements in figure 3:

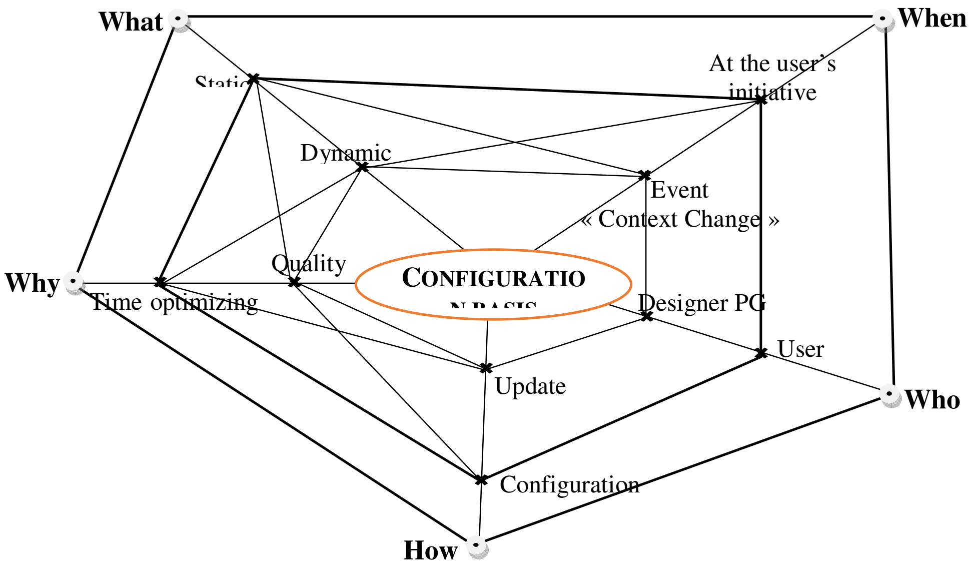

Figure 3. The configuration basis of guidance process.

This evolution basis allows, each time, to answer any manifest changes provoked statically or dynamically in order to design the most appropriate guidance process related to the current situation. The configuration management of guidance process is useful both for time optimizing and for the guidance process quality.

The (Re) configuration is invoked by the basic guidance process actors, namely the guidance process designer or at the user's request, with different interests ensuring an equilibrium between the user's needs in terms of guidance, with either static adaptation at the guidance conceptual level or dynamic process related to the development context change. The following illustration defines the basis description of the guidance invocation

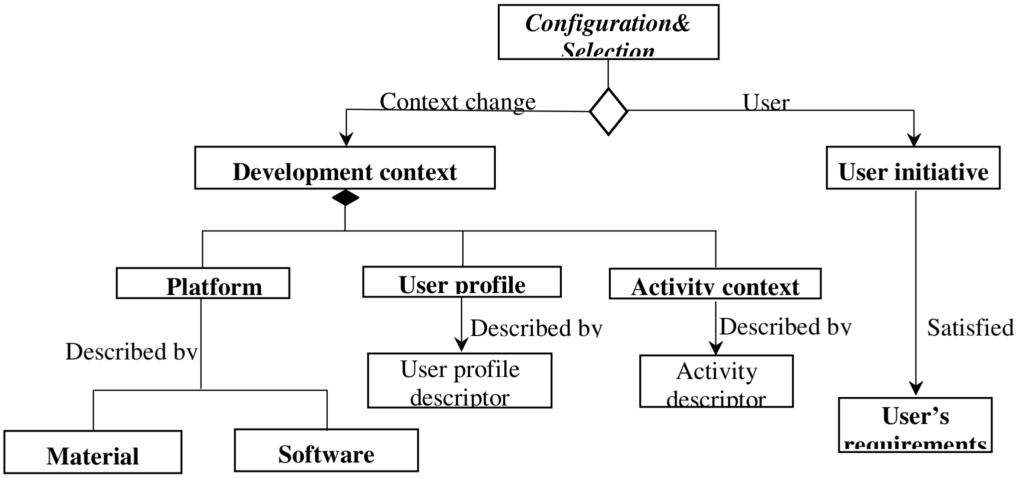

Figure 4. The basis description of the guidance invocation.

The involved configuration process deployment represents the core of the configuration management. It describes the steps and the contextual sequencing that should be undertaken to ensure this goal. Therefore, the deployment cycle of a guidance process configuration is explained via all activities related to the configuration management and their functional interdependencies (see figure 5.).

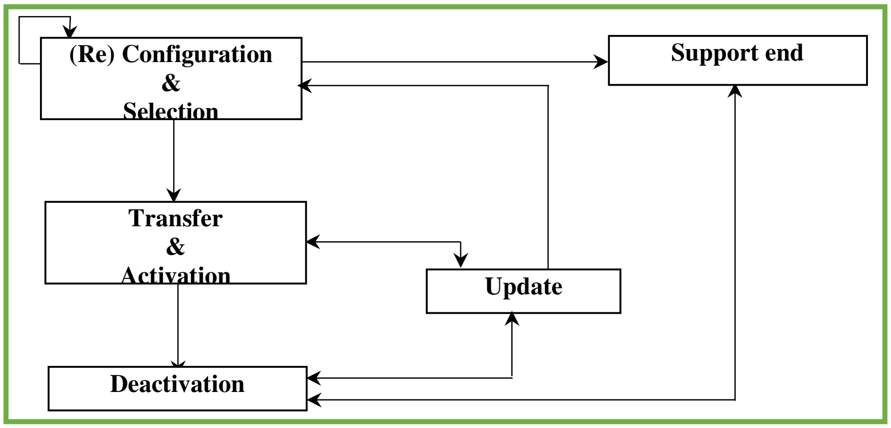

Figure 5. The guidance process configuration.

- The first activity concerns the configuration and selection of the most coherent composition, which best responds to the defined development context. Its objective is to offer the most appropriate guidance to support the current situation.
  For this, we should first deal with the dependencies between the guidance process and description of the development context, to specify the expected guidance service and constraints to be respected.
- The transfer and activation activity consists in transferring the guidance support. It is invoked at the initiative of the guidance process designer or at the request of the user. Once installation and transfer problems solved, the activation operation role is to put the guidance support in the active state, ready to be executed.
- The update activity is to resume, adjust or to complete the guidance support configuration. It is invoked at the user's initiative or by the guidance process designer following a change in the development context.

This activity is almost similar to the (re) configuration, with the difference that the guidance support is already installed. Finally, the implementation of the update uses the deactivation activity.

- The "support end" activity is relative to the end of guidance service offered to users of the software process in progress.
- Deactivation activity is essential before any configuration, update or support end activities. Its role is to deactivate the support guidance component so that we can perform the likely changes of some support components

These activities and their functional interdependencies give a rough estimation on the complexity of the of the guidance process configuration management. This process aims to set up, in respect of the development context constraints, a first core adaptation of guidance support. Subsequently, it will have to activate the dynamic adaptation operation with regard to the likely evolutions of the development environment.

4.1. THE CONFIGURATION OF GUIDANCE PROCESS META MODEL

Our modeling approach CGPM (***Configuration of Guidance Process Meta model***) is defined with reference to studied PSEEs features. The aim of our approach is to better respond to the factors of the development context defined by the three dimensions, the development context, adaptation form and offered guidance. Each dimension is described through several basic factors to develop a coherent configuration strategy and provide the most appropriate guidance support to the current development context.

In this context, our meta-model is based on a conceptual model of a software process enriched by a configuration model adapted to the guidance process. The configuration model detects any event of the development context evolution and launches the configuration of guidance process.
The role of the first activity "configuration and selection" is to prepare the guidance support to the current situation. The second service “transfer and activation” is to put the guidance support in the active state. The third service "Update" is to provide a way to modify and / or complete the guidance configuration following a development context change.

The configuration 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 execution in the development context, if required conditions related to the context and the adaptation form then launch guidance configuration strategy to generate the most appropriate guidance service.

The proposed meta-model aims at generating the adapted guidance interventions to the development context in relation to the considered factors and specific data for each defined dimension (see Figure 6)

.

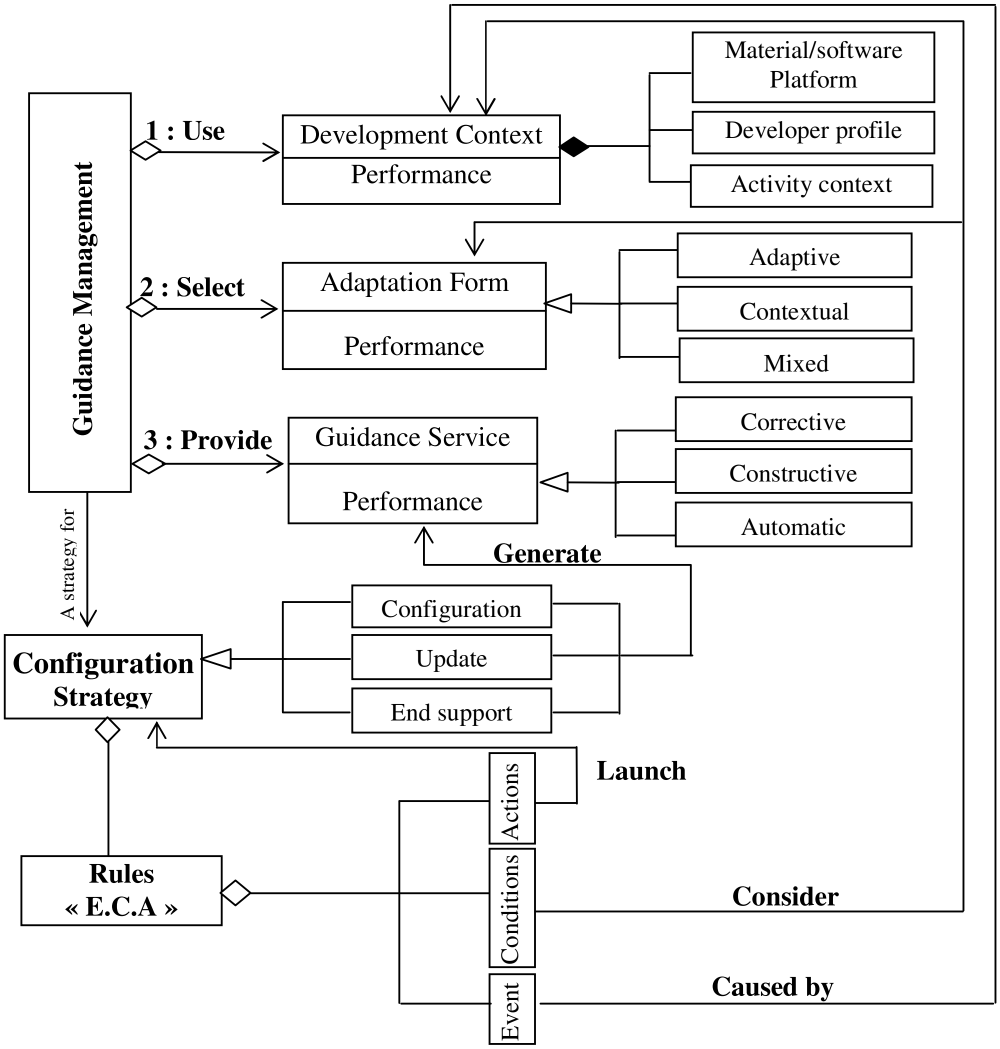

Figure 6. Configuration of Guidance Process Meta model.

5. 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.

A performing tree given in Figure 7 describes the activity process «Activity test». We notice that the activity test starts the execution of sub activities "Unitary test" then “Integration test". The unitary test launches in parallel the execution of tasks "Test unit"

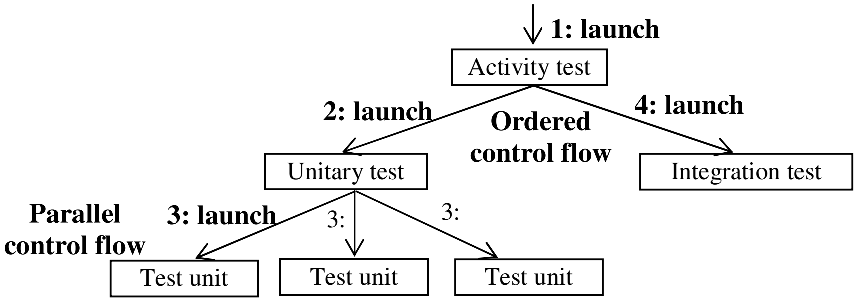

Figure 7. The activity test process.

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 in Figure 8.

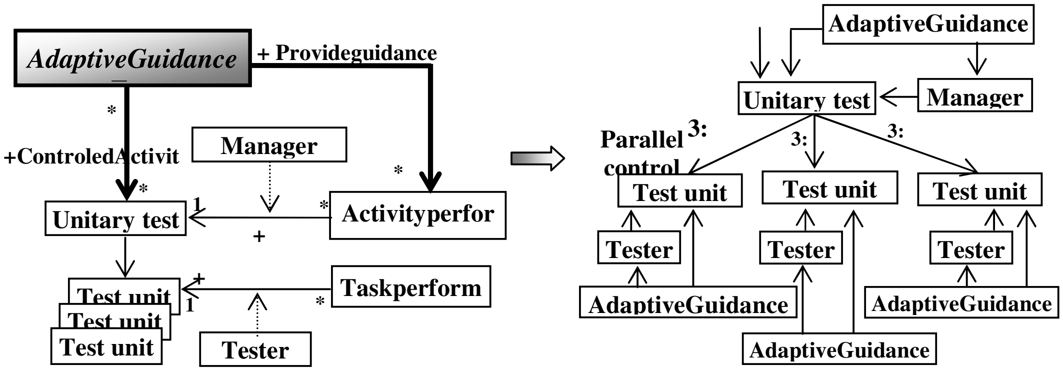

Figure 8. The adaptive execution process “ Unitary test”.

The adaptive guidance is linked to the manager or to each tester according to the current development context profile defined by its material/software platform, the activity context and developer profile. 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/software platform specificity differ from one development context to another. According to the current context, it would require 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/software 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; the manager should validate it.

3. For a development context with a low qualification tester and a reduced material/software 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.

5.1. The digital application

The practical definition of the adaptive guidance type for each considered profile is deduced through a quantitative process of the factors in relation to the basic models (material/software platform, activity context, developer profile). The considered example is processed as follows :
Each profile is semantically described in table (see Table 2). The project manager under the specification of an ongoing project [17] determines the semantic evaluation and the weighting. To scan the semantics evaluation, we associate the weighting related to the interest granted to each attribute.

Table 2. The profiles evaluation.

Development context	**Factors**	**Context Profile 1**	**Context Profile 2**	**Context Profile 3**	**Context Profile 4**	**Context Profile 5**	**W[i]**
Material/ software Platform	Development System Constraint	Low	Medium	High	High	Low	**P2**
	Software Tools	Low	Medium	Low	Provided	Provided	**P1**
	Memory Constraint	High	Medium	Medium	High	Medium	**P3**
Developer Profile	Role	No effect	Classic	Critique	Critique	No effect	**P4**
	Competence	High	Medium	Low	Low	High	**P1**
	Familiarity with Software Process	Quite Acceptable	Medium	Low	Low	Acceptable	**P1**
	Behavior for guidance	Most Appropriate	Satisfying	Inadequate	Adequate	Inadequate	**P2**
Activity Context	Density of tasks in the activity	Acceptable	Acceptable	Acceptable	Acceptable	Acceptable	**P3**
	Complexity Level	Medium	Medium	Medium	Medium	Medium	**P2**

With ***W[i]∈ [1, 5].*** Where Pi represents the computing value.

Considering the similar principle such as the COCOMO model [26], 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	Factors	Context Profile 1	Context Profile 2	Context Profile 3	Context Profile 4	Context Profile 5	W[i]
Material/ software Platform	Development System Constraint	**0.75**	**1.00**	**1.30**	**1.30**	**0.75**	**1**
	Software Tools	**1.20**	**1.00**	**1.25**	**0.80**	**0.80**	**3**
	Memory Constraint	**1.40**	**1.00**	**1.00**	**1.60**	**1.00**	**2**
Developer Profile	Role	**0.40**	**1.00**	**1.90**	**1.70**	**0.40**	**4**
	Competence	**0.20**	**1.00**	**1.70**	**1.70**	**0.25**	**3**
	Familiarity with Software Process	**0.40**	**1.00**	**1.60**	**1.60**	**0.30**	**3**
	Behavior for guidance	**0.20**	**0.80**	**1.70**	**0.75**	**1.60**	**1**
Activity Context	Density of tasks in the activity	**0.80**	**0.80**	**0.80**	**0.80**	**0.80**	**2**
	Complexity Level	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1**

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:

$GP\ (Px) = \Sigma\ Ai\ Wi\ /\ 2*\Sigma\ Wi$ ***avec i=1 to n***

With :

Ai: the feature value.
Wi: the associated weighting.
Px : 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.

	Context Profile 1	**Context Profile 2**	**Context Profile 3**	**Context Profile 4**	**Context Profile5**
Associated Adaptation form	Adaptive	Contextual	Mixed	Mixed	Adaptive
Guidance Profile (GP)	0.333	0.485	0.721	0.673	0.315
Associated guidance profile	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, a constructive guidance to profile P2 and automatic guidance to profile P3and 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.

6. CONCLUSION

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

The contribution of this approach is to institute the configuration management concepts, in order to be able to proceed, every time, with the adaptation of the guidance support to the various changes in the development context to ensure its adaptive deployment to software process.
The configuration strategy for the adaptation is done by the dynamic deployment of the provided guidance process. Intuitively, we consider the defined guidance process configuration through the sequencing of contextual activities. The guidance strategy evolves according to the political autonomy given to the guidance process 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 that allow swapping through different guidance profiles, either statically by adjusting the guidance parameters or dynamically through the development context changes.

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., Vanderdonckt, 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] C. Lueninghoener. "Getting Started with Configuration Management. ;login: issue: April 2011, Volume 36, Number 2" (PDF). Retrieved 2012-11-23.

[15] IEEE Std-828-2012 IEEE Standard for Configuration Management in Systems and Software Engineering. 2012. doi:10.1109/IEEESTD.2012.6170935. ISBN 978-0-7381-7232-3.

[16] Ali, U., Kidd, C. Barriers to effective configuration management. Application in a project context: An empirical investigation, International Journal of Project Management, 2013b; http://dx.doi.org/10.1016/j.ijproman.2013.06.005.

[17] Grambow, Gregor and Oberhauser, Roy and Reichert, Manfred (2011) Enabling Automatic Process-aware Collaboration Support in Software Engineering Projects. In: Selected Papers of the ICSOFT'11 Conference. Communications in Computer and Information Science(CCIS).

[18] 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.

[19] 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

[20] 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).

[21] Leif Singer, Fernando Figueira Filho, Margaret-Anne Storey (2014) Software engineering at the speed of light: how developers stay current using twitter. 36 th International Conference on Software Engineering, ICSE May 2014. Pages 211-221 ACM New York, USA 2014, ISBN: 978-1-4503-2756-5.

[22] International Standard ISO/IEC 12207 : (2008). Systems and software engineering -- Software life cycle processes.

[23] International Standard ISO 10007:2003 for Configuration Management

[24] IEEE Standard 828-2012 - for Configuration Management in Systems and Software Engineering

[25] Lindkvist, C., A. Stasis, and J. Whyte, Configuration Management in Complex Engineering Projects. Procedia CIRP, 2013, Volume 11(0): p. 173-176.

[26] Barry W. Boehm, Chris Abts, A. Winsor Brown, Sunita Chulani, Bradford K. Clark, Ellis Horowitz, Ray Madachy, Donald J. Reifer, Bert Steece, 2009. Software Cost Estimation with COCOMO II. Prentice Hall Edition, ISBN: 0137025769, 978013702576.

Authors' profile

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 Guidace.

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.