



HAL
open science

Methodological Framework to Guide the Development of Continual Evolution Methods

Ornela Çela, Mario Cortes-Cornax, Agnès Front, Dominique Rieu

► To cite this version:

Ornela Çela, Mario Cortes-Cornax, Agnès Front, Dominique Rieu. Methodological Framework to Guide the Development of Continual Evolution Methods. Paolo Giorgini; Barbara Weber. Advanced Information Systems Engineering. 31st International Conference, CAiSE 2019, Rome, Italy, June 3–7, 2019, Proceedings, 11483, Springer, pp.48-63, 2019, Lecture Notes in Computer Science, 978-3-030-21290-2. 10.1007/978-3-030-21290-2_4. hal-02059792

HAL Id: hal-02059792

<https://hal.science/hal-02059792>

Submitted on 22 Jun 2021

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.

Methodological Framework to Guide the Development of Continual Evolution Methods

Ornela Cela, Mario Cortes-Cornax, Agnès Front and Dominique Rieu

University of Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG, 38000 Grenoble, France

`firstname.lastname@univ-grenoble-alpes.fr`

Abstract. Companies live in a fast-changing environment imposing to constantly evolve in order to stay competitive. Such an evolution is carried out through continuous improvement cycles or radical changes often based on innovation that concern their products, their processes, their internal organization, etc. We refer to this situation as continual evolution. There are two implications of such continual evolution from our viewpoint: (a) the instillation of the “no end point” philosophy in organizations and (b) the use of methods based (1) on continual evolution cycles (by opposition to project-based approaches that have delimited budget and dates and) and, (2) on autonomous and collective implication of the organization’s actors. This article presents a methodological framework, called As-Is/As-If framework to support method engineers in handling such continual evolution. The framework offers a process model and a product meta-model that are both reusable instruments, aiming to guide the construction of continual evolution methods. The process model and product meta-model can be seen as prototypical examples to be adapted in each situation at hand using heuristics proposed as part of the framework. The usefulness of the framework is illustrated through two methods adaptations.

Keywords: Method engineering, continual evolution, framework

1 Introduction

Today, companies live in a fast changing environment that demands them to constantly evolve in order to stay competitive [1]. This need implies to continually embrace improvement cycles or radical changes often based on innovation concerning their products, their processes, their internal organization, their collaboration with external partners, etc. Different methodologies have been proposed in order to guide these improvement cycles with a focus on process efficiency (Lean) [2], product quality (Six Sigma) [2], innovativeness of products by changing the organization’s conception paradigm (Design thinking) [3] or the external knowledge use (open innovation) [4]. Despite their differences in focus, all the aforementioned methodologies consider the evolution cycle as a project, without taking into consideration that further improvement or innovation cycles might follow. To emphasize this constantly changing situation, we introduce the concept of continual evolution, which refers to the idea that evolution imposes multiple cycles of change,

as companies are constantly seeking for new opportunities to improve their competitiveness. This concept advocates that understanding the need for continual change must be instilled in the organization's culture [5]. This understanding will help in reducing the resistance toward the change and in facilitating its absorption by the organizational actors.

This holistic view of continual evolution, which we propose, has an impact on methods to be used for engineering the changes. Our view is that these methods must (1) be based on continual evolution cycles, (by opposition to project-based approaches that have delimited budget and dates) and (2) promote as much as possible the active and collective participation of actors of the organization, who will "take the method experts' place". For example, the process owner could replace the green belt consultant in Six Sigma and guide the process mapping. In this view, ludic and participative approaches are particularly adapted in order to lower the barrier of entry for the participants.

In this article, we propose a methodological framework, called As-Is/As-If framework to support method engineers in constructing or adapting methods matching the need of continual evolution. The framework has been constructed following a bottom-up approach that generalizes the concepts and phases proposed in continual evolution methods developed during research projects in which our team was involved from 2012 to 2018. It offers a process model and a product meta-model that are both reusable instruments to guide the method engineer in constructing or adapting methods for continual evolution. Both artefacts, can be seen as prototypical examples to be adapted for the situation at hand using heuristics that are proposed as part of the framework. The aforementioned adaptations produce a new method that we call the target method.

The paper is organized as follows: Section 2 provides an overview of our approach. Section 3 introduces the main concepts of the methodological framework and illustrates them with examples of two existing methods specifically dealing with continual evolution in two different domains. Section 4 presents the models of the framework (the product meta-model and process model). Section 5 introduces our position in comparison with related works. Finally, Section 6 concludes the article and introduces our future work perspectives.

2 Overview of the Approach

The As-Is/As-If framework offers a process model (expressed with the intentional MAP notation [6]) and a product meta-model (based on UML notation). MAP models are directed graphs with nodes representing intentions and labelled edges capturing strategies that model processes in a declarative manner. The MAP formalism allows the modeler to view a process at different levels of abstraction. Fig. 1 introduces the top-level map representing the general view of the framework process model and the core meta-classes of the product meta-model. The detailed view of the models using different abstraction levels will be developed in Section 4.

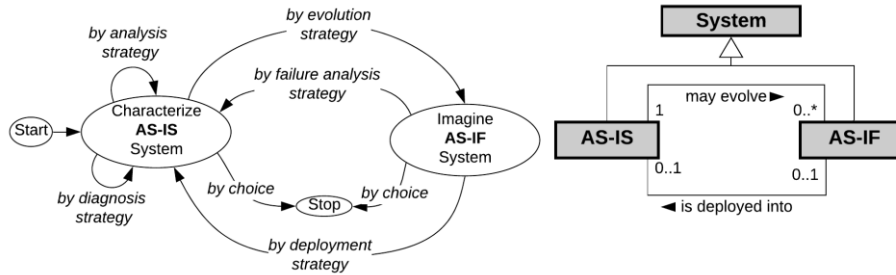


Fig. 1. General view of the process model and the product meta-model

As seen in Fig. 1, the **process model** has two main intentions *Characterize As-Is System* and *Imagine As-If system*, which must be achieved in this order. It shall be noted that we use the term system to denote the product that will be produced by the method. The framework is not dedicated to a specific type of product. Thus the use of the term system that can denote an information system, an ecosystem or a business process model. Each evolution cycle firstly consists in analyzing the current system (*by analysis strategy*) and performing a diagnosis (*by diagnosis strategy*). Once the first intention is attained, the goal is to imagine (*by evolution strategy*) possible evolution scenarios based on the question “*And if?*”. This strategy permits to determine a **set of possible As-If systems**, each of them corresponding to an evolution scenario. One of these multiple possibilities will be chosen and consolidated considering possible constraints (juridical, economic, social, technical, etc.). If no evolution scenario is chosen due to the impossibility to identify a satisfactory evolution (*by failure analysis strategy*), the analysis and the diagnosis strategies could be replayed. If an evolution scenario is chosen, it will be deployed becoming the new studied system (*by deployment strategy*). The decision to terminate the continual evolution cycle must be a collective choice of all actors. This possibility is represented through the *by choice* strategies. The framework **product meta-model** (right part of Fig.1) shows a simplified version of the associations between the As-Is and the As-If systems. It represents the concepts used by the process model.

The two models illustrate the top-level patterns provided by the framework to reason about the cycle of evolution and to help in decisions making when adapting them to the new situation at hand. We will see in Section 4 how the refinement of the top-level will progressively guide the adaptation of the framework; (1) process model to produce the process model of the target method and (2) of the product meta-model to complete the definition of the target method.

The framework generalizes our empirical experience in building continual evolution methods: ADInnov [7], ISEACAP [8] and CEFOP [9] methods are developed within research projects in which our team was involved. This paper relies on CEFOP and ADInnov, used as target methods to illustrate the usage of the framework in adapting both methods. The following section introduces the concepts manipulated by both methods that are generalized in the framework.

3 ADInnov and CEFOP: Concepts and Examples

This section introduces the main concepts of the ADInnov and CEFOP methods and how they are generalized in order to obtain the concepts of the As-Is/As-If framework. This presentation mimics the bottom-up approach we used to construct the framework product meta-model and process model. To facilitate the reading, the following rule is used: the generalized concepts (meta-concepts) of the As-Is/As-If framework are in **Bold**, the terms concerning the methods are in *bold italic*, and examples are in *italic* font. This section illustrates how the meta-concepts relate to either concept of ADInnov or of CEFOP. In addition, some examples of participative and ludic techniques used in these methods are provided as illustrations of concrete methodological fragments that a method engineer may reuse in the construction of a target method.

3.1 Concepts of ADInnov generalized in the As-Is/As-If Framework

ADInnov is the method resulting from the InnoServ project¹, which aimed to analyze and diagnose ecosystems formed by fragile persons at home and to propose innovation services in such ecosystems. This project was financed by the French national research agency (ANR) from 2012 to 2016. Here, the system under study (**As-Is System**) is the *ecosystem* around a fragile person at home.

The **Analysis** shows that this *ecosystem* is composed of different **Components**, partially represented in Fig. 2. For instance, the *target* is the legal or physical person (*fragile person*) benefiting from the ecosystem services and over which *actors* operate under their own business (*nurse, physician, etc.*). **Function** corresponds to a skill or responsibility in the ecosystem involved in the realization of a concrete service in the *ecosystem* (*health professionals*). Several **responsibility networks** are defined in order to manage the ecosystem complexity; they are views determined by their proximity with the *target*. Three **responsibility networks** (*RN*) were identified in the InnoServ project: *Regulation* (laws and rules concerning home care of fragile people), *Coordination* (home care organization of fragile people) and *Execution* (direct interaction with fragile people at home). A **point of view** (*financial, medical, social, etc.*) relates to a crosscutting vision that determines a point of interest of a provided **service**. Indeed, **services** concern the provision of technical and intellectual capacity or of useful work for a *target*. A **service** is attached to a *RN* and is composed of one or several **concrete services** treating a **point of view** and performed by one or more **functions** (the **service** *Recognize the caregiver work* can be done from a legal **point of view** *recognizing the caregiver status* and from a financial **point of view** *establishing a salary for caregivers*). **Measures** such as the *quality of life of fragile people* or the *time to go back home after hospitalization* evaluate the *ecosystem*.

The **Diagnosis** of the ecosystem reveals several **Blocking Points**. In the context of the InnoServ project, the same term **blocking point** is used, which corresponds to a concrete issue in the context of a **responsibility network** or a **point of view** (in the

¹ <https://goo.gl/8HHZYQ>

Execution RN: there is a problem of unavailability as well as lack of required actors for caregiving). A **point of view** is translated into one or more refined **Goals** (have available actors in the fragile person's house). **Constraints** are used to express the fact that some components can't evolve (the fragile person is not supposed to evolve).

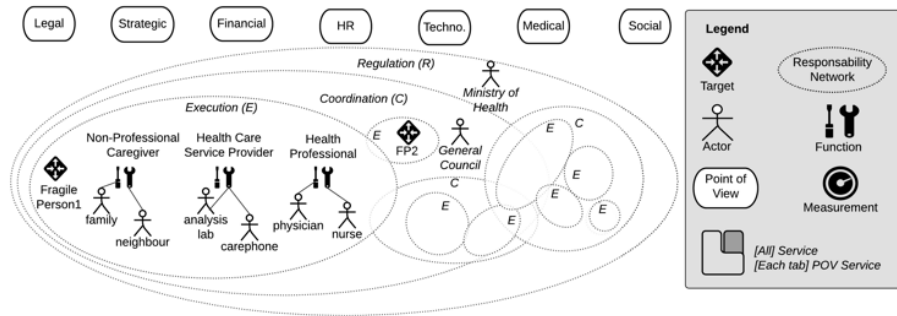


Fig. 2. Components of the As-Is ecosystem resulting from Analysis in the ADInnov method

The **Innovation** phase aims to propose several **Changes** that make possible to reach the identified **Goals**. In ADInnov, **Changes** are named *innovation services* such as the proposition of new *services* (a digital service of piloting) or new *functions* (orchestrator, coordinator) [7]. The impact of these *innovation services* lead to **Evaluation** in order to identify their impact. The *innovation services* will introduce one or more *point of view service* (**Operational Changes**) corresponding to the implementation in the **As-Is system**. They can affect one or more **Components**: modifying them, deleting or adding new ones. For example, it is easy to imagine that the *innovation services* proposed above will have a great impact on the territory organization that will have to be at their turn, analyzed and diagnosed, hence considered as new *As-Is ecosystem*.

In order to come with the *ecosystem's components* (**Components**), detect the *blocking points* (**Blocking Points**) or propose *innovation services* (**Changes**) in the *ecosystem* (**System**), different strategies are used, which are considered here as reusable method fragments. Some of these method fragments such as serious games¹ to analyze the information flow between *functions*, post-its to study service dependency or CAUTIC workshops [10] to evaluate *innovation services* were proposed [7] and could be re-used in any other target method with similar needs.

3.2 Concepts of CEFOP generalized in the As-Is/As-If Framework

The CEFOP method [9] is one of the results of ProMiNi², a research project in collaboration with NetInvaders, a French web development agency specialized in e-shop solutions and thin client software. The goal of this project was to apply a process mining approach to analyze the traces left in the company's information system in

¹ <http://www.lego.com/fr-fr/seriousplay/>

² <https://en.net-invaders.com/>

order to evolve its business processes to stay competitive in the fast changing environment. Here, the **As-Is System** is the *business process*.

The **Analysis** of the process (*the client's subscription process*) aims to elicit the **Components** (see Fig. 3): *inputs, outputs, tasks, actions, roles and tools*. The *inputs* (*Subscription request*) and *outputs* (*Successful subscription or Rejected subscription*) delimit the process under study. The *tasks* describe the activities performed in the process related to one *role* participating in the process (*client, etc.*). *Tasks* are composed of *actions* (*Create Account*), each of them describing an atomic activity supported by at most one *tool* (*mysql*). The performance of the process is evaluated by *indicators* (*the maximal number of days to subscribe*) that correspond to **Measure Elements** computed over the traces produced during the process execution. A set of indicators is defined relying on *goals* and *constraints*, elicited during the diagnosis.

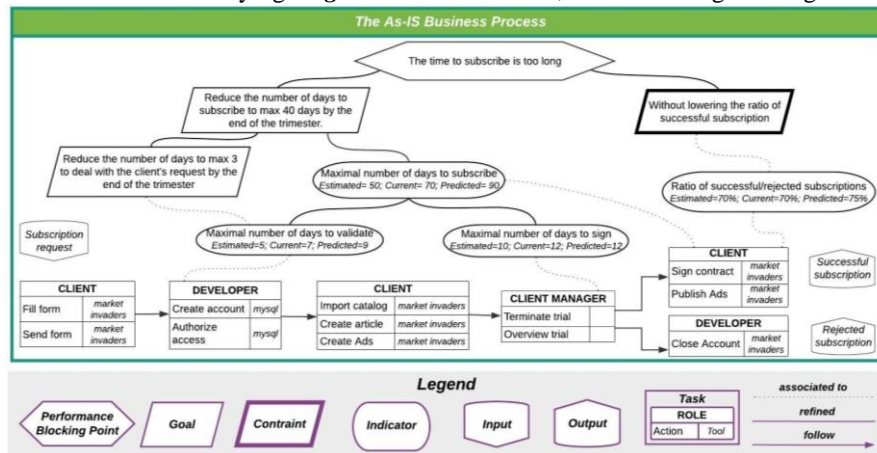


Fig. 3. Components of the As-Is business process elicited in the CEFOP method

The **Diagnosis** of a *business process* focusses on identifying the *performance blocking points* (*The time taken to subscribe is too long*). They describe performance elements currently missing to be introduced in the evolution cycle. **Blocking Points** are translated into *goals* expressing the desired results to be achieved (*Reduce the number of days to subscribe to max 40 days by the end of the trimester*) and *constraints* that avoid changing specific values (*Without lowering the ratio of successful subscription*).

The **Evolution** phase starts by identifying possible **Changes** and translating them into **Operational Changes**. Each of them describes how the components of the **As-Is** are affected when generating the different **As-Ifs**. Indeed, several possible ways of evolving a process, implementing different changes, can be proposed (*As-If processes*). For example, a possible evolution (*As-If1*) for *the client's subscription process* could be generated by taking into consideration a *change* (*Hire an additional client manager*), implying two *operational changes* (*Increment the number of actors playing the client manager role and Recalculate the indicators*). In order to choose the most appropriate way to evolve the business process, the potential evolutions are

subject to **Evaluation**. This implies to check if the *performance blocking points* are resolved, the *goals* are achieved without violating the *constraints*, and the steps to be deployed are defined.

When eliciting the above elements, CEFOP strongly requests the process participants in order to reduce the need of external intervention. For that, the method integrates several ludic and participative methodological fragments [11].

3.3 Construction of the As-Is/As-If Product Meta-model

Fig. 4 summarizes the As-Is/As-If product meta-model as resulting of the generalizations of specific concepts into generic ones that we illustrated in the previous sections. Models of the specific projects (Innoserv and ProMiNi concepts) are instantiations of the ADInnov and CEFOP concepts (the meta-classes of the product meta-models). The method concepts are then generalized to produce the As-Is/As-If framework concepts, which are detailed in the next section.

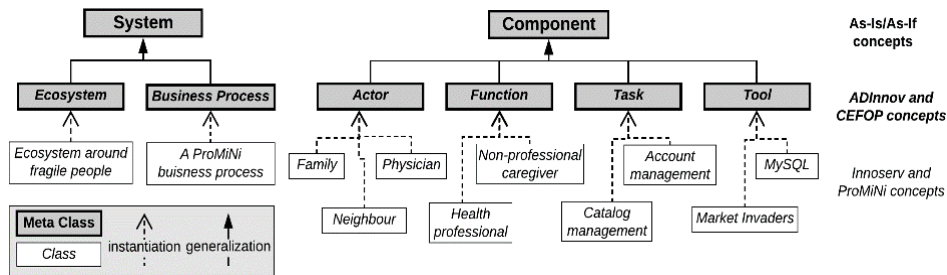


Fig. 4. Construction of the As-Is/As-If product meta-model

4 As-Is / As-If Methodological Framework

This section presents in detail the As-Is/As-If framework and illustrates its use to adapt the initial versions of the CEFOP and ADInnov methods. In this section we use a top-down approach to review the CEFOP and ADInnov methods using the framework. During this process, we discovered imperfections in these two methods. Thus, the use of the framework for these methods resulted in a second version of the product meta-model and process model for both methods as illustrated in the following subsections.

4.1 General View

According to the OMG modeling levels [13], a method is composed of a product meta-model (level M2) that defines the abstract syntax of the modeling language and a process model (level M1) whose execution produces product models that conform to the product meta-model. The As-Is/As-If framework (Fig. 5) proposes two models that can be reused and adapted to formalize continual evolution methods. The **process model** (left part) is formalized using intentional maps [6] and a **product meta-model** (right part) is formalized in a UML class diagram.

The process model defines the main strategies and intentions of any continual evolution process model and provide the starting point for eliciting the elements of the product meta-model. It is represented through a general map and three sub-maps that refine the Analysis, Diagnosis and Evolution sections of the top level map. Accordingly, the product meta-model is structured in 4 packages: the Core and three packages corresponding to Analysis, Diagnosis and Evolution.

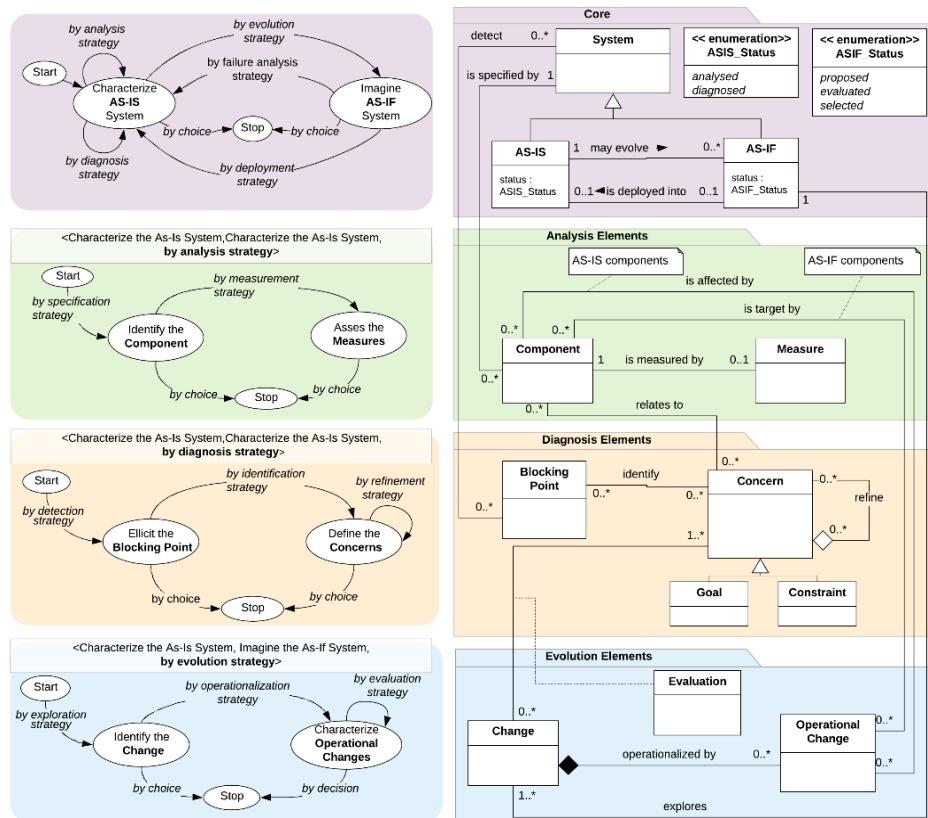


Fig. 5. The process model and the product meta-model proposed in the As-Is/As-If framework

The core package is composed of 3 mandatory meta-classes: the **System** meta-class and its two subclasses. An **As-Is System** can evolve into several possible evolution scenarios called **As-If Systems** (*may evolve* association). As mentioned before, only one **As-If System** will become the next **As-Is System** (*deployed into* association). The status attributes represent the **System State** with respect to its evolution process. An **As-Is System** can be **analyzed** (associated with elements of the Analysis Element package) or **diagnosed** (associated with the Diagnosis Element package). Also, the **As-If System** state can be **proposed** (associated to the Evolution Element package), **evaluated** (if an evaluation between the elements of Evolution and the Diagnosis exist) and **selected** (**As-If system** is deployed into an **As-Is System**).

It shall be noted that association naming provides the model-level traceability. For instance, the *indicator- maximal number of days to subscribe* of the *As-Is process* corresponds to the same *indicator* in the *As-If process* but it has different *values*, since the *activities* are affected by the *change*.

The product meta-model and the process model are closely related since the intentions of the process model define the elements of the product meta-model, so:

1. Each **meta-class** in the product meta-model relates to an intention of the process model. The name of the intention is composed by the name of the meta-class and a verb describing the operation to be performed (identify, elicit, characterize, etc.)
2. The **associations** between the meta-classes correspond to strategies to be followed in order to attain an intention. These strategies are named using the structure “by noun strategy” proposed in the MAP formalism [6]. The “noun” derives from the name of the association in the product meta-model.

Both the process model and the product meta-model provide the base that a method engineer will further enrich while constructing or evolving a continual evolution method. Regarding the process model, maps offer a starting point to be adapted and refined using the following heuristics:

1. **Intentions** can be renamed, but no intention can be deleted or added.
2. The proposed **strategies** illustrate paths to be followed. They can be renamed but not deleted. If required, new strategies between existing intentions can be added.
3. The **sections** of the analysis, diagnosis and evolution sub maps can be refined in order to detail how components, blocking points, changes, etc. of the target method are identified and characterized.

Concerning the product meta-model, the adaptations and refinements heuristics are:

1. The **core elements** cannot be extended.
2. The **meta-classes** can be renamed but no class can be deleted or added. The meta-classes of the analysis, diagnosis and evolution packages can be extended into more detailed class diagrams.
3. The **associations** can be renamed and new ones can be added.

4.2 Adaptation of the Product Meta-model

Undertaking continual evolution implies to adapt the product meta-model. This is guided by the use of the heuristics introduced in the above section. Fig. 6 illustrates such an adaptation for two different target situations, namely the ones corresponding to the ADInnov and CEFOP methods (version 2). The Core package is not illustrated since it cannot be changed. The new associations identified when adapting the product meta-model are illustrated in thicker lines. Bold named meta-classes illustrate renaming or decomposition with regard to the framework. The languages corresponding to ADInnov and CEFOP are therefore defined by the abstract syntax given by the product meta-model in Fig. 6, and a concrete syntax introduced in the legends of the examples of sections 3.1 and 3.2, respectively.

The Analysis package essentially captures the **Components** of the system being studied and the **Measures** used to evaluate them. The **Component** meta-class is maintained but extended in the form of a class diagram containing all the components of the system. The refinement of this class permits the method engineer to adapt the method to a specific domain. The framework recommends that a model should never describe the entire system. Instead, it should be limited to the components targeted by the continual evolution. For instance, the CEFOP method does not take into account the synchronization of the control flows and therefore gateways are not considered as **Components**. During the continuous evolution process, new **Components** may appear, in particular in the case of disruptive evolution or innovation. In ADInnov, the concept of *responsibility network* (a new type of component) appeared in the first iteration cycle. **Components** are evaluated through **Measures**. For example, CEFOP uses *indicators* to measure the execution time of *activities*.

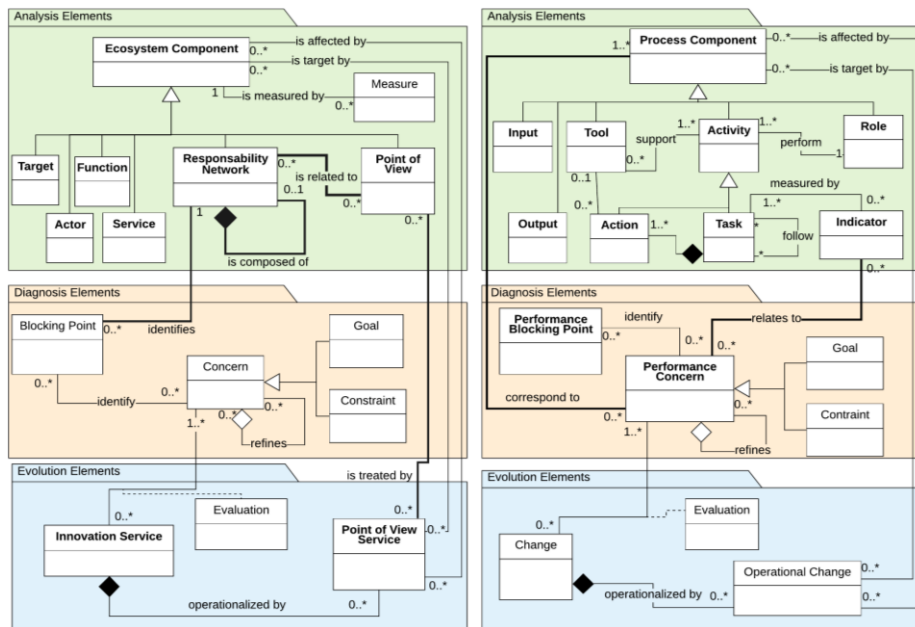


Fig. 6. Target methods adaptation corresponding to the abstract syntax of ADInnov and CEFOP

The Diagnosis package has two main meta-classes: **Blocking Point** and **Concern**. The former refers to a specific problem in the system that needs to be resolved and can be expressed as a simple sentence, a text or set of basic attributes. In a target method, the **Blocking Point** meta-class must be included in order to synthesize in a simple way the reason of an evolution. **Concern** is an abstract class that can be either a **Goal** or a **Constraint**. The target method must identify **Goals** to be reached and potential **Constraints** to be preserved in order to solve the corresponding **Blocking Point**. Concerning CEFOP, in order to resolve the *performance blocking points* (the time taken to subscribe is too long), the *goals* (Reduce the number of days to subscribe [...] trimester) must be achieved and the *constraints* (without lowering the

ratio of successful subscription) preserved. CEFOP associate *goals* and *constraints* with the *process components* (correspond to association) to better conform to the studied domain.

The Evolution package proposes three meta-classes: **Change**, **Evaluation** and **Operational Change**. A **Change** is evaluated in relation to **Concerns**. Their implementation is detailed by **Operational Changes** that describe how **As-Is Components** are *affected* in order to generate the *targeted As-If Components*. For instance, in CEFOP a change was *operationalised* by two *operational changes* in order to describe how the possible evolution could be obtained.

4.3 Adaptation of the Process Model

This section derives the process model of a target method by adapting the As-Is/As-If framework process model. A target method should conform the top-level map by keeping the same strategies and possibly renaming the intentions. For example, in ADInnov and CEFOP, **System** is replaced by *ecosystem* and *business process*, respectively. As illustrations of adaptation we show the adaptation of the analysis sub map for CEFOP and the one of the evolution sub map for ADInnov.

The **Analysis** sub map illustrated in Fig. 5 proposed two main intentions: **Identify the Components** of the system and **Assess the Measures**. The framework advocates to start **by specification strategy** and then to reach the second intention **by measurement strategy**. The **by choice** strategy provides some flexibility if required to swap from analysis to diagnosis. CEFOP uses the same sub map, renaming the framework intentions to *Identify the process components* and *Measure performance* (see Fig. 7.a). Additionally, all the sections of the analysis sub map are further refined. Fig. 7.b illustrates the refinement of the *section* <Start, Identify the process components, by specification strategy>. Two intentions are considered: *Outline the process*, reached **by analysis strategy** and *Model the process* reached **by human process modelling strategy** or **by automatic process discover strategy**. The former strategy aims to model the process through the ISEAsy tool [12] and the latter aims to discover the model through the usage of process mining techniques. The last strategy in this section is **by model consolidation strategy**, which serves to consolidate deviations between the human modelling and the process mining.

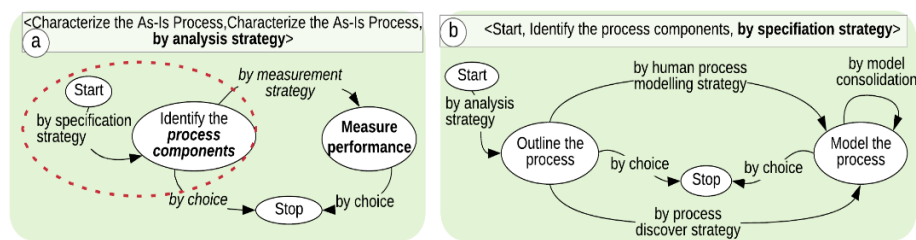


Fig. 7. The analysis map adapted by CEFOP and the refinement of the by specification section.

The adaptation of the **Evolution** strategy is illustrated for ADInnov. The main intentions to be achieved are the *Identify innovation services* and *Characterize innovations*. In the Evolution map (Fig 8.a), we just renamed the intentions to better

fit the ADInnov domain and we added a new strategy *by enrichment strategy* that go back from the characterization to the identification. The second map in Fig. 8.b refines the section <Characterize innovations, Characterize innovations, by evaluation strategy>. The intentions are *Consolidate the innovations* as well as *Illustrate the innovations* (by creating several videos illustrating the proposed innovations¹). Strategies of this section are for instance, the use of serious games, the identification of service dependencies (using post-its) or an evaluation workshop using the CAUTIC method. More details about these re-usable method fragments can be found in [7].

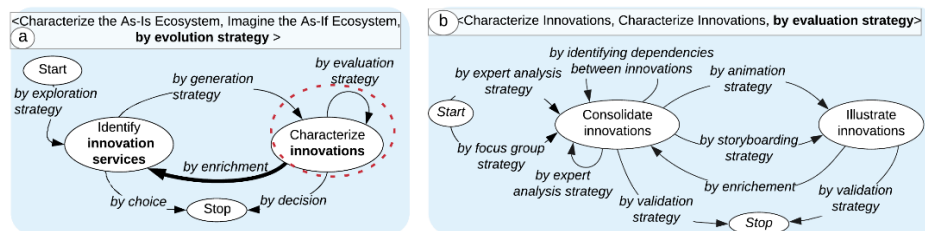


Fig. 8. The evolution map adapted by ADInnov and the refinement of the by evaluation section

5 Related work

The As-Is/As-If framework belongs to the **Method Engineering** discipline, defined as the discipline to design, construct and adapt methods, techniques and tools for the development of information systems [14]. However, the framework is not limited to methods dealing with the development of information systems. It is wider in terms of target, since the type of studied system can be an ecosystem (ADInnov), a business processes (CEFOP), or any type of artefact. Vice-versa, the framework is not as generic as a typical method engineering solution would be as it focuses on the specific domain of continual evolution. Indeed, it proposes models (a process model and a product meta-model) dedicated to a specific method family [15], which is the family of continual evolution methods.

Different types of approaches are proposed in ME to deal with the construction of a new method by instantiation or adaptation of models [14-18]. The As-Is/As-If framework adopts the latter with specificity: the product meta-model and the process model of the target methods are at the same abstraction level as the framework models. These models have to be adapted in order to define the product meta-model (the abstract syntax of the language) and the process model of the target method. The product meta-model of the As-Is/As-If is therefore an instance of the MOF and does not benefit from the modeling power of UML. To solve this issue, OMG proposes the UML profile mechanism to reduce the time and the cost of meta-model development [19]. The As-Is/As-If meta-classes would be UML stereotypes and the meta-classes of the target product meta-model are themselves stereotypes extending As-Is/As-If stereotypes (in UML, a stereotype can specialize another stereotype). The adaptation thus consists in using the profile mechanism to define the abstract syntax of the target product meta-model.

¹ <https://anrinnoserv.wordpress.com/>

The product meta-model cannot be used without adaptation and thus does not provide a concrete syntax. When adapting the product meta-model, a concrete syntax (e.g., a graphical notation) associated with the most specialized level of stereotypes and corresponding to the concepts of the target method must be defined. In ADInnov, the profile approach is particularly interesting since it provides the possibility to define a Domain Specific Language dedicated to ecosystems and to introduce appropriate extensions for health ecosystems.

To compare our proposal with **methods dealing with changes** we first consider the PDCA cycle [20] (*Plan Do Check Act*) which is a four steps method that provides the foundations upon which most of continual improvement methods constructed their cycles. *Plan* studies the current way of doing and identifies possible evolutions for improving the process. *Do* corresponds to the realization of the improvements identified in the previous step. *Check* overviews the impact of the evolution and checks if the desired results are obtained. Finally, *Act* operationalizes the improvement if the previous check was positive. Our framework focuses on the coverage of the *Plan* phase through the As-Is/As-If cycle. To this intent, the *Plan* phase of the following methods has been analyzed. The deployment strategy implies going over the *Do*, *Check* and *Act* phases to obtain the new As-Is system.

The Lean cycle [21] is composed of 5 steps: *Value- Value stream mapping - Flow - Pull- Seek perfection*. The first two phases could be included in the PDCA *Plan*. They aim to detect the blocking points describing the problem as well as the client's needs and identifying the process and its wastage's causes. These steps are followed by the process improvement and monitoring and the trigger of further improvements if required. The DMAIC (*Define-Measure-Analyze-Improve-Control*) cycle [22], used by Six Sigma, correlates with the PDCA *Plan* in its three first phases: *Define* (aiming to identify the project scope and the resources composing the project), *Measure* (evaluating the current performance of the system) and *Analyze* (diagnosing performance problems and identifying their root causes).

Aligning the plan phase of the PDCA cycle with the previous methodologies is natural since their aim is continual improvement. This alignment is less evident when analyzing innovation methods. A comparison of the *Build-Measure-Learn* cycle of Lean Startup and the *Empathize-Define-Ideate-Prototype-Test* cycle of Design Thinking has been done [23]. The authors highlight that even through a different number of phases and names, their cycles aim to achieve similar goals. In particular, the *Learn* phase in Lean Startup has the same intention than the Design Thinking *Define* step. They both outline the context, elicit the clients and their requirements relating them to the *Plan* phase in the PDCA, as intended by our framework.

6 Conclusion and Perspectives

This paper proposes a methodological framework that guides the formalization of methods that support continual evolution. It provides a product meta-model and a process model to be adapted by method engineers to deal with a specific evolution domain. An important goal is that the target methods could be provided to organizations and be used in autonomy, minimizing the need of a method expert. The

main force of the framework is the As-If dimension allowing constructing As-Ifs scenarios and evaluating their impact. The framework benefited from our experience in building continual evolution methods in different contexts and different research projects. Nevertheless, we are aware that the framework still needs to be tested in different contexts, evaluated, improved and enforced on several points.

First, a complete methodological guide is necessary to help the method engineers in adapting the product meta-model and the process model. The product meta-model will be completed with a dictionary to propose definitions, attributes and examples of each concept. Moreover, considering the strategies of the process model, guidance on the reuse of the aforementioned methodological fragments supported by participative, elicitation or creativity techniques, needs to be enhanced. We are currently working on solutions to guide the method engineer in choosing the best methodological fragment adapted to her/his specific context. One promising solution is the development of software tools supporting the library of methodological fragments deployed on MethodForChange¹, a platform aiming to collect and provide methods and tools related to innovation.

Second, in terms of evaluation and scaling up, more experiments are necessary. The framework has been used in adapting, generating new versions of three methods, which is insufficient, but new projects in our research team will allow us to test the As-Is/As-If framework. In the context of the IDEX CIRCULAR project² (2018-2021), a method to facilitate the transition towards circular economy in industrial supply chains will be proposed. In the AURA MOBIPA research project (2018-2021), a method will aim to improve the access of elderly people to mobility services. These two experiments will allow us to enrich the As-Is/As-If framework and support their construction. Indeed, the framework is easily extendable, adding new associations in the product meta-model and new strategies in the process model.

Acknowledgments. We sincerely thank Prof. C. Rolland for her valuable feedbacks.

References

1. Rashid, O. A., Ahmad, M. N.: Business Process Improvement Methodologies : An Overview. In: Journal of Information Systems Research and Innovation (2013), vol. 5, pp. 45-53
2. Singh, J. Singh, H.: Continuous improvement philosophy– literature review and directions. IN: Benchmarking: An International Journal (2015), vol. 22 (1), pp. 75-119
3. Brown, T., and Wyatt, J.: Design thinking for social innovation. Development Outreach (2010), vol.12 (1), pp. 29-43
4. Chesbrough, H., & Bogers, M. Explicating open innovation: Clarifying an emerging paradigm for understanding innovation. New Frontiers in Open Innovation. Oxford University Press, Forthcoming (2014), pp. 3-28.
5. Georgiades, S.: Employee Engagement and Organizational Change. In: Employee Engagement in Media Management (2015), Springer, pp. 19-37

¹ <https://methodforchange.com/>

² <https://circular.univ-grenoble-alpes.fr/en/main-menu/circular/>

6. Rolland, C.: Capturing System Intentionality with Maps. In: *Conceptual Modelling Information System Engineering*, Springer-Verlag (2007), pp. 141–158
7. Cortes-Cornax, M., Front, A., Rieu, D., Verdier, C., Forest, F.: ADInnov: An Intentional Method to Instil Innovation in Socio-Technical Ecosystems. In: *International Conference on Advance Information System Engineering (CAiSE 2016)*, Springer, pp. 133–148
8. Movahedian, F., Front, A., Rieu, D., Farastier, A. *et al.*: A participative method for knowledge elicitation in collaborative innovation projects. In : *International Conference on Research Challenges in Information Science (RCIS 2017)*, IEEE, pp. 244–254
9. Cela, O., Front, A., Rieu, D.: CEFOP: A method for the Continual Evolution of Organisational Processes. *International Conference on Research Challenges in Information Science (RCIS 2017)*, IEEE, pp. 33–43
10. Forest, F. Mallein, P. Arhippainen, L.: Paradoxical User Acceptance of Ambient Intelligent Systems: : Sociology of User Experience Approach. In: *Proceedings of International Conference on Making Sense of Converging Media (2013)*, ACM, pp. 211–218
11. Çela, O. Front, A. Rieu, D.: Model Consolidation: A Process Modelling Method Combining Process Mining and Business Process Modelling. In: *International Conference on Business Process Modeling, Development and Support, BPMDS (2018)*, Springer, pp. 117–130
12. Front A., Rieu, D. Santorum, M. Movahedian, F.: A participative end-user method for multi-perspective business process elicitation and improvement. In: *Software System Modelling (2017)*, vol. 16 (3), pp. 691–714
13. OMG: Meta Object Facility (MOF) version 2.5 (2016). Available on: <https://www.omg.org/spec/MOF>
14. Brinkkemper, S.: Method Engineering: Engineering of Information Systems Development Methods and Tools. In: *Information and Software Technology (1996)*, ScienceDirect, vol. 38(4), 275–280
15. Deneckere, R., Kornysheva, E., Rolland, C.: Method Family Description and Configuration. In: *International Conference on Enterprise Information Systems (ICEIS 2011)*, SCITEPRESS, pp. 384-387
16. Ralyté, J., Rolland, C., Ben Ayed, M.: An Approach for Evolution Driven Method Engineering. In: *Information Modeling Methods and Methodologies (2005)*, IGI Global, pp. 80-101
17. Hug, C., Front, A., Rieu, D., Henderson-Sellers, B.: A Method to build Information Systems Engineering Process Metamodels. In *Journal of Systems and Software (2009)*, Elsevier, vol. 82(10), pp. 1730-1742
18. Cervera, M., Albert, M., Torres, V., Pelechano, V.: A Methodological Framework and Software Infrastructure for the Construction of Software Production Methods. In: *International Conference on Software Process (ICSP 2010)*, Springer, pp.112-125
19. Lakhil, F., Dubois, H., Rieu, D.: Pattern Based Methodology for UML profiles Evolution Management. In: *IEEE - International Conference on Research Challenges in Information Science (RCIS 2013)*, pp. 1-12
20. Deming, W. E.: *Out of the crisis*. In: Cambridge, Massachusetts Institute of Technology (MIT), Center for Advanced Engineering Study (1986)
21. Womack, J. and Jones, D. T.: *Lean Thinking: Banish Waste and Create Wealth for Your Corporation*. In: *Journal of the Operational Research Society (1996)*, vol. 48 (11), pp. 1148–1148
22. Hambleton, L.: Define-Measure-Analyze- Improve-Control (DMAIC). In: *Treasure Chest Six Sigma Growth Methods, Tools, Best Practices (2007)*, pp. 13–27
23. Mueller, R. M. Thoring, K.: Design Thinking Vs Lean Startup: A Comparison of Two User driven Innovation Strategies. In: *Leading through Design, (2012)*, vol.151, pp. 91-106