Dynamic adaptation of business process models
Application to the healthcare process in AP-HM

Renaud ANGLES1,2, Philippe RAMADOUR2, Corine CAUVET2, Sophie RODIER1
1 Assistance Publique – Hôpitaux de Marseille
Direction des Systèmes d’Information et de l’Organisation
147, Boulevard Baïlle
13 005 MARSEILLE, FRANCE
2 AMU (Aix-Marseille University), LSIS (UMR CNRS 7 296)
Domaine Universitaire de Saint-Jérôme
Avenue Escadrille Normandie-Niemen
13 397 MARSEILLE cedex 20, FRANCE
{firstname.lastname}@lsis.org, http://www.lsis.org

Abstract— Healthcare organizations, which are facing the challenge of delivery personalized services to their patients, are obviously affected by the problems of flexibility and adaptability of their processes. This research is applied to healthcare processes in the context of AP-HM hospitals (Assistance Publique - Hôpitaux de Marseille). In this paper, we consider specifically the drug circulation process where the complexity and the high level of variability are critical issues and important in practice. The paper introduces the V-BPMI approach for process variability and it presents how dynamic adaptation can be carried out for delivering process models that satisfy actor’s business requirements. The paper focusses on both the steps of the adaptation cycle and the adaptation trees dynamically produced on business actors’ demand.

Keywords—Process flexibility, Process adaptation, Adaptation trees, Variability trees

I. INTRODUCTION

Companies have identified enterprise information systems agility as a competitive advantage required for increasing product and service customization, for improving quality of products and services delivered and for adapting their business rules to highly dynamic working environments. Healthcare organizations, which are facing the challenge of delivery personalized services to their patients, are obviously affected by the problems of flexibility and adaptability of their processes. Many reports in the healthcare field state that there is an “absence of real progress towards applying advances in information technology to improve administrative and clinical processes” [1]. Furthermore, in healthcare organizations, the lack of flexibility of enterprise information systems is considered as a major obstacle in improvement of organizational and medical treatment processes.

This research is applied to healthcare processes in the context of AP-HM hospitals (Assistance Publique - Hôpitaux de Marseille). In this paper, we consider specifically the drug circulation process where the complexity and the high level of variability are critical issues and important in practice.

Our research is based upon a recent information system paradigm known as PAIS (Process-aware Information Systems): a PAIS is defined as “a software system that manages and executes operational processes involving people, applications and/or information sources on the basis of process models”. Flexibility requirement for PAIS therefore raises two issues: (i) how to express and manage variability in process models at design-time [2], [3] and (ii) how to take into account the business environment for adapting business process models at run-time [1], [4], [5].

The proposal advocates V-BPMI, an approach where process models emphasize variability and are supported by services. V-BPMI provides a process modeling language (so-called V-BPMN). Goal and context are the main concepts introduced to support process variability and adaptation. Process models are memorized in a process repository statically structured with arborescent links (variability trees). Dynamic adaptation of process models consists in discovering and composing available process models to satisfy actor’s business requirements. The paper focusses on dynamic adaptation of process models. It introduces both the steps of the adaptation method and the adaptation trees dynamically produced on business actors’ demand.

The remainder of the paper is organized as follows. In section 2, we introduce the requirements of variability and adaptability in the drug circulation process of AP-HM hospitals that have motivated this research. Section 3 presents an overview of V-BPMI approach. Section 4 introduces the V-BPML base. Section 5 describes the dynamic adaptation method and explains dynamic adaptation trees for producing context-dependent process models. Section 6 presents operators supporting dynamic adaptation in V-BPML. Section 7 presents related work and section 8 concludes this paper.

II. PROCESS VARIABILITY REQUIREMENTS FOR THE DRUG CIRCULATION IN THE AP-HM

This section introduces the drug circuit process in the AP-HM organization and it highlights the highly dynamic environment of this process. Such a process requires a very flexible approach to adapt its execution on the fly in order to deal with constraints in front of which this process is executed.

A. Overview of the drug circuit process

The process of circulation of the drugs is complex. But, at high level, it is generally accepted that it can be specified with three phases as shown in Fig. 1.
The prescription is performed by a doctor in a medical unit, according to a diagnostic. The dispensation consists in the preparation and the delivery of the prescribed drugs. Pharmacists are responsible for validating prescribed drugs and carrying out the preparations in the pharmacy. Nurses are in charge of the administration phase, so they are responsible for giving the adequate drugs and monitor the patients.

B. Process variability

If we consider now in details the sub-process prescription, it is a loosely specified process which has to be refined by end-users during run-time, for example taking into account that if the doctor is a senior the prescription is send immediately, whereas if he is a junior the prescription has to be validated by a senior. In this example, there is a predefined constraint leading to execute or not some validation activities. In practice, due to the high number of choices, not all of them can be anticipated and hence pre-defined in a unique process model.

The AP-HM organization manages 4 different hospitals with their own pharmacy. Each of them is concerned with the dispensation phase of the drug circulation process. However, due to available resources which differ from one pharmacy to another, each one performs a specific variant of the drug circuit process to satisfy the same business requirement.

Most of healthcare processes are complex and they are partially realized by existing legacy applications which can be shared among different processes. In addition, the benefits of process automation from within a single hospital can be transferred to other hospitals. Moreover, some process activities are similar in all cases and there are some differences regarding the involved software components. Nowadays, the service paradigm seems to be a further step in process flexibility due to “late bidding” possibility of services. So, services registries must be defined, managed and maintained.

C. Drug circuit process and its working environment

In practice, there are a large variety of constraints which impact process definition and deployment as shown in Fig. 2. The law around the drug circulation process is highly fluctuative and revisions of the way the process has to be run are often required, so legal constraints impact processes. The pharmacy size and the available storage space have an impact on the storage policy. Every constraint linked to the resources and the environment is considered as an environmental constraint. It is possible to perform a task with different strategies: the storage policy or the period of the day devoted to the dispensation, for example, are organizational constraints. To finish, the same technologies are not available in every pharmacy: the Wi-Fi coverage or the used software are not the same for example. These are technical constraints.

The constraints where a process is deployed influences the way it has to be modeled and the way it will be run [6], [7].

Fig. 1. High level BPMN description of the drug circulation process

Our proposal introduces a methodology, V-BPMI, to model and manage process variability. Taking into account the specificity of each deployment context, V-BPMI provides a dynamic adaptation approach to produce a process suitable to this context. In this paper, we only consider a simplified representation of the dispensation phase of the drug circulation process to illustrate V-BPMI.

III. V-BPMI OVERVIEW

The V-BPMI approach introduces concepts to model and produce flexible processes in alignment with the business requirements. This section presents the architecture of V-BPMI and the main conceptual tools supporting it (Fig. 3).

V-BPMI is mainly supported by:

- A V-BPMI base: this repository contains V-BPMI process lines and process variants, which support process variability modelling and dynamic production of adapted processes (cf. IV and V).

- A services repository: due to the services orientation of processes in V-BPMI, the service repository is used to implement such processes.

- A domain ontology: this ontology is used both for the production and the usage of V-BPMI concepts (process
lines and process variants). This domain ontology contains the goals, contexts, actors, processes, resources of the domain and ontological links. It also contains the domain terms with semantic links (mainly synonymy, paronymy, hypernymy and hyponymy). The Fig. 3 underscores the central role of the domain ontology.

- A V-BPMI adaptation engine: this is the core of the V-BPMI adaptation approach. This engine is used by business actors (in our case, healthcare actors) to produce dynamically adapted processes.

The V-BPMI approach adopts a dual orientation:

- This is an intentional approach: the notion of goal is one of the main concepts supporting V-BPMI. Goals allows to define variable processes, thus information systems, in alignment with the strategy of the enterprise. According to this orientation, we consider that the deployment of a business process allows satisfying a business goal.

- This is a contextual approach: due to the process constraints interaction (cf. Fig. 2), we consider that a goal can be satisfied in several ways, depending on the situation in which it has to be satisfied. The contextual orientation is powerful for describing several processes satisfying the same goal, each one being discriminated by the context in which its deployment is the more relevant.

IV. THE V-BPMI BASE

We define the V-BPMI approach to model, store and manage flexible processes. According to the dual orientation of V-BPMI, we consider that a business process satisfies a goal in a relevant specific context.

There are several languages for business process modeling. One of the most common is BPMN [8]. This language mainly allows expressing activities and their scheduling. Despite the notion of ad-hoc processes, BPMN unfortunately doesn’t focus on the variability. That’s why V-BPMI introduces the V-BPMN language, which encapsulates BPMN and allows modeling new concepts introduced in V-BPMI. One of the reasons of the choice of a language encapsulating BPMN is the service approach for the operationalization of the processes. It introduces a first level of flexibility, allowing choosing the way to operationalize a BPMN service task with a “late binding” possibility. An advantage of this choice is that a BPMN process is also a V-BPMN process.

This section presents the different concepts related to the variability before introducing the V-BPMI base used to store the variable processes.

A. V-BPMI Concepts Supporting the Variability

V-BPMI introduces some concepts for the process variability modeling and management. We describe here the main V-BPMI concepts supporting this dual orientation.

- A goal allows describing the finality of a business process. Its expression is based on domain ontology. Goals introduce a way for supporting the alignment between the strategy of the enterprise and the process deployed. Inspired by [9], we propose to formally express a goal with an action and an object concerned by the action: \( (\text{To do})_{\text{Action}} (\text{Something})_{\text{Object}} \)

For example, \( (\text{Pick up})_{\text{Action}} (\text{Drugs})_{\text{Object}} \) is an healthcare goal.

- A context is the formal expression of specific situation in which the deployment of a process is relevant. A context is a set of contextual assertions supported by the domain ontology. The assertions are typed and logically linked in a context with an AND operator (a context is a conjunction of contextual assertions). Some of them can be negated with a NOT operator.

For example, \( (\text{Dispensation type} = \text{emergency})_{\text{Environmental}} \text{AND} (\text{Storage mode} = \text{Robot})_{\text{Resource}} \) is an healthcare context.

- A process line abstracts all the ways (i.e. business processes) for satisfying a business goal. A process line is identified by a business goal. Each business goal to be satisfied in the domain is associated with a process line in the V-BPMI base. Thus, in a V-BPMI base, all the business processes satisfying a business goal \( G \) are associated with the process line identified by \( G \). (cf. Fig. 4).

Fig. 4. V-BPMN notation of the collapsed view of the process line of the drug dispensation

- A process variant contains the description of a business process in which the variability can be emphasized (thus, this is a V-BPMN process, as shown in Fig. 5). So, a process variant provides one of the ways for satisfying a business goal in a specific context.

Fig. 5. V-BPMN notation of the expanded view of a process variant

A process variant is always associated with a process line: the one identified by the goal satisfied by the V-BPMN process contained in the process variant. Several process variants can then be associated with the same process line, each one
providing a V-BPMN process satisfying the goal that identifies the process line. That’s why we discriminate each process variant by a context which identifies it (cf. Fig. 6).

The V-BPMN process included in a process variant can contain one or more references to process lines. Such references allow expressing that a business requirement has to be satisfied here and it can be satisfied in several ways. This leads us to identify 2 types of process variants:

- **Operationalizable process variants** contain no reference to any process line. In this case, the business process included in the process variant is an usual BPMN process which can be operationalized (for example in BPEL [10], [11]).

- **Abstract process variants** contain at least one reference to a process line. In this case, the business process included in the process variant is a V-BPMN process that can’t be immediately operationalized: a choice has to be made to select a specific way for satisfying the business goal which identifies each referenced process line. The example of process variant in Fig. 5 is an abstract process variant: the V-BPMN process it contains refers 2 process lines.

The language V-BPMN defines 2 symbols to identify operationalizable process variants and abstract process variants (cf. Fig. 7).

![Fig. 6. V-BPMN notation of the expended view of the process line of the drug dispensation associated with 2 collapsed process variants](image)

**Fig. 6.** V-BPMN notation of the expended view of the process line of the drug dispensation associated with 2 collapsed process variants

**B. Structure of the V-BPMI base**

The concepts introduced above allow producing flexible processes models. It is important to store the models in a base taking care on the flexibility and the contextual and intentional approach.

A process line and its associated process variants can be structured in a two-level tree. The process line is the tree root and they are as many leaves in the trees as process variants, either operationalizable or abstract, associated with the process line. The link between the process line and the process variants is a selection link (i.e. an XOR link). This kind of tree is called a variability tree in the V-BPMI approach.

![Fig. 7. Collapsed V-BPMI notations for operationalizable process variants and abstract process variants](image)

**Fig. 7.** Collapsed V-BPMI notations for operationalizable process variants and abstract process variants

![Fig. 8. An example of a variability tree](image)

**Fig. 8.** An example of a variability tree

Leaves of the variability trees are process variants. Thus, they can be operationalizable or abstract. Let’s remember that operationalizable process variants are BPMN processes whereas abstract process variants are V-BPMN processes in which at least one reference to a process line appears.

So, due to abstract process variants, which can reference process lines, it can be interesting to dynamically link some of the variability trees. This is the role of dynamic adaptation trees presented below.

**V. DYNAMIC ADAPTATION OF PROCESSES**

We introduce in this section the concept of dynamic adaptation trees and their usage during the production of adapted processes. This production is conducting according with the cycle of dynamic adaptation.

**A. Dynamic adaptation trees**

The abstract variants associated with a process line refer to other process lines contained in the V-BPMI base. Thus, it is possible to dynamically link an abstract process variant to the process lines it references. This can be done by linking the variability tree of which the process variant is a leaf with the variability tree of which the referenced process line is the root.

For example, the process line (Dispense)Action (Drug)Object presented in Fig. 6 is the root of a variability tree which has 2 leaves corresponding to the 2 abstract process variants appearing in Fig. 6. One of these variants, which is detailed in Fig. 5, refers 2 process lines: (Validate)Action (Prescription)Object.
and \((\text{Pick up})_{\text{Action}} (\text{Drugs})_{\text{Object}}\). It means that, to operationalize this process variant, both referenced process lines have to be satisfied. Thus, it is possible to dynamically link the process variant of the process line \((\text{Dispense})_{\text{Action}} (\text{Drug})_{\text{Object}}\) relevant in the context \((\text{Dispensation type} = \text{emergency})_{\text{Environmental}}\) with the process line \((\text{Validate})_{\text{Action}} (\text{Prescription})_{\text{Object}}\) and the process line \((\text{Pick up})_{\text{Action}} (\text{Drugs})_{\text{Object}}\).

This kind of link is a **dynamic composition link** (i.e. an AND link). Fig. 10 illustrates that link.

With such dynamic composition links, and existing static selection links structuring the V-BPMI base, it is possible to compose variability trees. The result of the composition of variability trees is called a **dynamic adaptation tree**. The root of an adaptation tree is a process line and the leaves are process variants. Nodes of odd levels are process lines while nodes of even levels are process variants (either operationalizable or abstract). The links from process lines of an odd level \(n\) to process variants of the even level \(n+1\) are XOR links (selection links). This links are static and are those which structure the V-BPMI base throughout the variability trees. The links from the process variants of an even level \(n\) to process lines of the odd level \(n+1\) are AND links (composition links). This links are dynamic.

The **research** phase requires both a process line and a context specification. The V-BPMI adaptation engine determines the more relevant process line, according to the domain ontology. If there is no process line satisfying this goal, the user can dynamically create one which then will be stored in the V-BPMI base.

For example, when the preceding business requirement is expressed, the adaptation engine will search in the V-BPMI base the more relevant process line. If the domain ontology defines the terms “Dispense” and “Give” as synonyms, the engine will select the variability tree (cf. Fig. 12) associated with the process line introduced in Fig. 4.

**B. Cycle of dynamic adaptation**

This cycle of dynamic adaptation is triggered when a business requirement is expressed. A business requirement is formally structured with a business goal (the one to be satisfied) and a business context (the one in which the goal has to be satisfied).

For example, a healthcare business requirement can be:

\[
((\text{Give})_{\text{Action}} (\text{Drug})_{\text{Object}})_{\text{Goal}} \land ((\text{Dispensation type} = \text{emergency})_{\text{Environmental}} \land \ldots \land (\text{Storage mode} = \text{Robot})_{\text{Resources}})_{\text{Context}}
\]

The output of the cycle of dynamic adaptation is an operationalizable BPMN process which satisfies the goal of the business requirement in the expressed context.

The cycle of dynamic adaptation is made of 4 phases.

For example, in our example, the more relevant variant in front of the initial business requirement is the second one (on the right). Thus, the right-branch of the variability tree is selected, as shown in Fig. 13.
The operationalization phase aims at specifying the way of operationalizing all of the BPMN activities of the selected variant. For example, for BPMN service tasks, the binding of services has to be done. The adaptation engine will check in the service base an adequate service for each BPMN service task.

In our example, the selected process variant contains a BPMN service task (identified by “Select the drug request”). This service task can be bind with a specific web service, as shown in Fig. 14.

The composition phase depends on the type of the current variant. If the variant is operationalizable, it contains a usual BPMN process which has been operationalized in the previous phase. It then can be translated in a BPEL process (for example) which is executable. Thus, in this case, the composition phase is omitted.

If the variant is abstract, it then contains at least one reference to process lines. Such references have to be operationalized. This is the objective of the composition phase. This stage aims at composing variability trees, which results in a dynamic adaptation tree:

- The one in which the abstract process variant appears as a leaf: it will be at the top of the dynamic adaptation tree resulting from the composition,
- The ones in which the referenced process lines appear as roots: they will be sub-trees in the dynamic adaptation tree resulting from the composition.

The sub-trees have to be produced with new iterations in the cycle of adaptation: input of these new iterations is a business requirement expressed as following: the goal is the one identifying a referenced process line and the context is the actual context.

Thus, recursively, a dynamic adaptation tree is produced. The iterations are stopped when all leaves of the dynamic adaptation tree are operationalizable process variants, i.e. when whole the variability has been “frozen”. All BPMN processes appearing in all leaves are then composed and the result is a classic BPMN process which can be translated in a BPEL process [10] to be executed.

In our example, the abstract process variant refers to 2 process lines, each one associated with a variability tree:

- The first referenced process line, identified by the business goal (Validate)Action (Prescription)Object, corresponds to the variability tree shown in Fig. 15.
- The second referenced process line, identified by the business goal (Pick up)Action (Drugs)Object, corresponds to the variability tree shown in Fig. 8.

For each of those referenced process lines, a new iteration has to be done in the cycle of dynamic adaptation, i.e. research of the more relevant variability trees, selection of a unique branch in those trees, operationalization of the produced V-BPMN process and, possibly, composition with other variability trees.

In our example, the final dynamic adaptation tree produced is illustrated in Fig. 16. In this figure, we partly show it: some of the unselected branches don’t appear. The selected branches are shown in bold.
All leaves of this dynamic adaptation tree are operationalizable process variants, which then contain BPMN processes. These BPMN processes can be composed according to the selected branches of the dynamic adaptation tree. This results in a BPMN business process satisfying the goal of the initial business requirement and relevant in the context specified in this business requirement.

Dynamic adaptation trees are conceptual tools supporting the cycle of dynamic adaptation. This cycle is also supported by a set of operators described in the next section.

VI. OPERATORS SUPPORTING DYNAMIC ADAPTATION

We define a set of operators inspired of [12] supporting the production of dynamic adaptation trees, i.e. the cycle of dynamic adaptation. Three classes of operators are defined: ontological operators for terms and contextual assertions equivalence evaluation, similarity operator defined on goals and compatibility operator defined on contexts and adaptation cycle operators (selection, composition...).

A. Ontological equivalence operators

These operators support comparison between terms and comparison between assertions by exploiting semantic links (mainly synonymy, paronymy, hypernymy and hyponymy) defined in the domain ontology.

The operator evaluates the rate of semantic equivalence between 2 terms or 2 groups of terms and . The result of this operator is a float in [0..1] which corresponds to the rate of semantic equivalence between and calculated as a distance, in the domain ontology, throughout semantic links between and .

The operator is an operator which defines the rate of semantic equivalence between contextual assertions. Let’s remember that a contextual assertion is typed and has a formulation. Thus, we can formally express a contextual assertion with a couple , where is the type of the contextual assertion and is its formulation. Let’s consider two assertions and . The rate of semantic equivalence between and is null (0) if otherwise (the operator is the same as presented before). Thus, the result of this operator is a float in [0..1] which corresponds to the rate of the semantic equivalence between and if and 0 otherwise.

B. Similarity and compatibility operators

These operators are mainly defined for matching goal and context within a business requirement:

- goalsSimilarity(): the goal similarity is evaluated as follows: the goal is composed of Action and . Then, the similarity between and is calculated as follows: similarity = (Action and ) × (Object and Object).
- contextsCompatibility(): the context compatibility is described as following: we define Pos as the set of the assertions contained in and which are not operand of a NOT operator. We define Neg as the set of the assertions contained in and which are operand of a NOT operator. Pos and Neg are defined in the same way. Then, the compatibility between and is calculated as fallowing:

\[ \forall A \in Pos(C_1), \exists A' \in Pos(C_2) \mid (A \equiv A') \quad \text{return } P, \]
\[ \forall A \in Neg(C_1), \exists A' \in Neg(C_2) \mid (A \equiv A') \quad \text{return } N, \]

\[ \text{compatibility}(C_1, C_2) = [P \times \overline{N}], \]

C. Adaptation cycle operators

We introduce here 5 operators: the first one supports whole the cycle of dynamic adaptation, and the other ones support the 4 stages of this cycle.

- adaptationCycleIteration(businessRequirementGoal G, Context C): this operator implements an algorithm triggering iteration(s) of the adaptation cycle. Its input is a business requirement and its output is a V-BPMN process satisfying the goal in the context C. This operator is based on the 4 next ones.

- processLineResearch(Goal G): input of this operator is a goal G. It researches in the V-BPMN base all process lines identified by a goal similar to the goal G. If several process lines are returned, they can be ordered by similarity value with the goal G. Output of this operator is in fact a variability trees corresponding to the process line identified by the goal G’ the most similar to the goal G. This operator is used during the research phase of the cycle of dynamic adaptation.

- processVariantSelection(Context C, Variability Tree VT): inputs are the variability tree VT produced beforehand and the context C of the business requirement. It research in the process variants of VT the one discriminated by the context C’ the most compatible with the context C. This process variant is called PV, has a type (operationalizable or abstract) and contains a V-BPMN process Proc. This operator is used during the selection phase of the cycle of dynamic adaptation.

- processVariantOperationnalization(processVariant PV): input is a process variant PV, which has a type (operationalizable or abstract) and contains a V-BPMN process Proc. For each activity in Proc marked as a service task (according to BPMN definition), bind a convenient service referenced in the service repository. This operator is used during the operationalization phase of the cycle of dynamic adaptation.

- processVariantComposition(processVariant PV): input of this operator is a process variant PV, which has a type (operationalizable or abstract) and contains a V-BPMN process Proc. If PV is operationalizable, then this operator returns Proc, which is a BPMN process. If PV is abstract, then the following algorithm is executed:
This set of operators supports all the phases of the cycle of dynamic adaptation. They allow a business actor to express a business requirement and get back a BPMN process in which service tasks are bind with available web services.

VII. RELATED WORK

Several approaches address variability in process modeling [13], [14]. These approaches often consider variability capture in process models. C-EPC [15] is an extension of the language EPC and of the ARIS method [16]. It introduces the notion of configurable nodes, configurable functions and the guidelines to support the flexibility of the processes. PROVOP [17] starts with a generic model that contains some adjustment points to identify the variability zone in the process. It is possible to define some sets of actions called options to modify (add, delete, or modify) the activities to build an adapted process. BPCN [18], [19] is a hybrid approach, blending a declarative and descriptive definition of the process. There is a static part of the process, and an ad-hoc part. In this part some non-scheduled available activities are defined. BPCN introduce two kinds of constraints networks to describe the way to use the non-scheduled available activities. The notion of constraints network is used in the DECLARE/YAWL framework [20], [21]. The language DECLARE permits to describe a process only with sets of constraints. They can be mandatory or optional according to the need. It is possible to define constraints templates to aggregate some sets of constraints under a conceptual high level constraint.

Even if the existing methods propose powerful concepts for variability capture they consider a little the intention and the context of a process. In [22] the authors propose to link a context and a goal to every process version, and in [23] it is possible to link an intention to a process description to support the variability of the organizational dimension of the process. These approaches are concerned with process variability modeling and they little exploit goal and context for guiding process adaptation. In [24], the authors consider run time adaptation by allowing the user to modify the process model. Process adaptation guidance and process adaptation automating are yet research issues. V-BPMI dynamic adaptation cycle based on process lines and process variants reuse is a step further in process adaptation.

VIII. CONCLUSION

We introduce the V-BPMI approach to model process variability and provide tools and methodology for contextualized processes production. The V-BPMI adaptation cycle allows selecting process variants and composing the relevant process lines in order to construct a process satisfying a business requirement. Dynamic adaptation trees and operators have been defined to support the adaptation cycle.

We actually address the definition of an architecture for V-BPMI implementation. This architecture is service-oriented: in particular, it involves user’s interface services, ontology services (for the manipulation of the domain ontology) and services for management of the V-BPMI base.

In the future, dynamic adaptation trees should be used at design-time to evaluate the consistency of the V-BPMI base and help the process designer in process lines and process variants production.

In the AP-HM context, memorizing dynamic adaptation trees should be an interesting issue for traceability of the drug process design.

IX. BIBLIOGRAPHY


