International Journal of Computer and Electrical Engineering, Vol. 2, No. 1, February, 2010
1793-8163

Modeling and Analysis of Change Management in Dynamic Business Process

Bassam Atieh Rajabi, Sai Peck Lee

Abstract—**Many business processes are highly dynamic and require changes even during execution. Existing commercial Business Process Management Systems (BPMS) fail to support such processes appropriately since they work in a rather static manner; they demand that the structure of a process is fixed before execution. The aim of this research is to provide a change management technique to improve the flexibility, adaptability, and dynamic of the current Business Process Modeling (BPM) approaches, and to help software engineers by finding a systematic and methodical approach for change analysis and management.**

Index Terms—**Business Process Modeling, Change Management, Graph Based Modeling, Rule Based Modeling. Object Oriented Modeling, Coloured Petri Net.**

I. Introduction

Software change management is an essential discipline for enterprise Information Technology (IT) organizations. In modern enterprises, software automates a wide variety of business processes. business process management technology continues to face challenges in coping with dynamic business environments where requirements and goals are constantly changing and thus business users are demanding adaptive and flexible frameworks for process management [1] [2]. Software change management is an essential discipline for enterprise IT organizations. In modern enterprises, software automates a wide variety of business process. Changes made to software are, in effect, changes made to the business processes themselves. Software change thus requires careful management. Without proper software change management, enterprises lack a full understanding of how software running in production automates their business processes [3]. The change according to [4] includes effects associated with the Strategy, Structure, System, Style, Staff, Shared Value (or subordinate goals), and Skill.

According to [1] the change impact could be Local (concerns on instances. It is necessary to detect if it can have indirect impacts on other instances) or Global (concerns on the process definition). Also the nature of the change could be Ad hoc (change dynamically performed on one or several instances when the process definition is not convenient for the execution conditions. The ad hoc change has thus a local impact), or Corrective (the change aims to correct a design error on the process definition or to react to an exception which happens during the execution of an instance. It can have local or global impacts), or Evolutionary (the change is required due to the redesign or reconfiguration of business processes. The old process definition is then considered as inappropriate with regard to the new management objectives. The evolutionary change has a global impact).

Existing BPMS distinguish between design time and run time modeling. The most important criteria for characterizing the functionality of control flow capabilities, considering the runtime requirements are: Flexibility, Adaptability, and Dynamic [1] [5] [6]. Change occurs frequently in current business processes due to two primary reasons [7]: the specification at design time is incomplete due to lack of knowledge, and errors or exceptional situations can occur during execution. These usually cause breakdowns, reduced quality of services, and inconsistencies.

In order to achieve flexible business processes, and make quick response to varying requirements, it is extremely important to propagate the change to the runtime instances [8].

In this research, we aim to design a framework for change management technique in BPM based on the integration between Unified Modeling Language (UML) diagrams as the Object-Oriented (OO) diagramming technique and Coloured Petri Net (CPN) as a Petri Net (PN) modeling techniques. OO technology has become extremely popular because of its provision of powerful structuring facilities, which stress encapsulation and promote software reuse, but it remains informal and still lacks tools for automatic validation [9]. Petri Nets (PNs) are used for the formal specification of concurrent systems. They have a natural graphical representation, which aids in the understanding of such formal specifications, and a range of automated and semi-automated analysis techniques, but the weakness of PN formalisms is the inadequate support for compositionality, which means providing the structuring facilities, encapsulation and inheritance [9]. The integration between OO and PN formalisms is very important to reap the complementary benefits of these two paradigms.

The rest of the paper is organized as follows: in Sections II a review for the BPM is provided, change management in BPM research issues are discussed in Section III. Observations from related researches published are given in

Bassam Atieh Rajabi , Software Engineering Department, Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. Email: bassam_rajabi@perdana.um.edu.my.

Sai Peck Lee, Software Engineering Department, Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. Email: saipeck@um.edu.my.

Section IV. In Section V, the framework development is presented. Finally, we conclude the paper and the future work.

II. Review of Business Process Modeling Languages (BPML)

BPML provides appropriate syntax and semantics to precisely specify business process requirements, in order to support automated process verification, validation, simulation and process automation. The syntax of the language provides grammar to specify objects and their dependencies of the business process, often represented as a language specific process model, while the semantics defines consistent interpretation for the process model to reflect the underlying process logic [6]. Graph based and rule based are two most predominant formalisms on BPML development.

A. *Rule Based Modeling Languages*

Rule Based Language is often referred to as Business Rule Management System (BRMS). The common objective of BRMS is to integrate complex process logic into a process model to support dynamic changes [6] [10]. In a typical rule based modeling language, process logic is abstracted into a set of rules, each of which is associated with one or more business activities, specifying properties of the activity such as the pre and post conditions of execution [6]. There are several classification schemas for business rules. According to [11], there are four kinds of business rules: constraint rule, action enabler rule, computation rule, and inference rule. Fuzzy business rules were added later as described in [12]. The following are examples for the rule based modeling languages:

Event Driven Process Chain (EPC) [13] [14]: the basic elements of it are functions and events. Functions model the activities of a business process, while events are created by processing functions or by actors outside of the model. Integrated Event driven Process Chains (iEPCs) basically extend EPCs with formal concepts of object flow and a role perspective [15]. The main idea is to show how any of these formalizations can be enhanced with transition rules that consider object existence and role availability as part of the state concept.

PLMflow [16] provides a set of business inference rules which is designed to dynamically generate and execute workflows. The process definition is specified in business rule templates, which include backward chain rules and forward chain rules. The process instance schema is determined by the rule engine using backward chain and forward chain inference at runtime. ADEPT system [17] is an infrastructure for designing and implementing multi-agent systems for workflows. Process logic is expressed in the service definition language, which specifies services that give the agents sufficient freedom to take alternative execution paths at runtime to complete the process goal.

ConDec language [18] is a declarative language to specify which tasks are possible. Users can execute such a model in their own preference, they can choose which tasks to execute and how many times, and in which order to execute tasks. Integration of Fuzzy aspects in business process management was developed by [12]. It extends the process modeling through the consideration and processing of fuzziness using the fuzzy-set-theory. This fuzzy extension will be reproduced with the EPC. It was shown that many situations in business process management could be described more exactly through the modeling of vague knowledge with fuzzy logic.

B. *Graph Based Modeling Languages*

In a graph based modeling language, process definition is specified in graphical process models, where activities are represented as nodes, and control flow and data dependencies between activities as arcs. The graphical process models provide explicit specification for process requirements [6]. The following are examples for the graph based modeling languages:

UML 2.0 Activity Diagram [19] [20] is designed for modeling business process and flows in software systems, and to provide a high-level means of modeling dynamic system behavior [20]. UML is a language for specifying, visualizing, constructing and documenting the artifacts of OO software systems, as well as for business modeling. UML uses OO methods for modeling [20] [21]. It is considered the standard OO modeling language.

Business Process Definition Meta-model (BPDM) [19]: The BPDM does not provide its own graphical notation, which is specified as a UML 2.0 profile. The intention of the BPDM is to define a generic meta-model in order to support the mapping between different tools and languages. Business Process Modeling Notation (BPMN) [22] is designed for modeling business process and their transformation into an execution language, namely the BPML [23]. The main concepts of BPMN are similar to UML 2.0 activity diagram.

OO Methodology is an established technique for structured software design. It supports Inheritance, Polymorphism, and Dynamic Binding. It is useful to designing software that is comprehensible, maintainable, and flexible [21].One of the main advantages of OO method is the effectiveness of the process to identify and refine objects [24]. OO is based on objects that represent real-world entities. Techniques for OO analysis and design primarily support the representation and integration of static system properties from a function and data perspective. Dynamic properties are only partially supported from a process perspective [25].

PN is a powerful instrument for modeling, analyzing, and simulating dynamic systems with concurrent and non-deterministic behavior. They are often criticized because their application for real-world problems leads to incomprehensible and complex net models [25]. It is useful in describing information systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic and/or stochastic. The graphical representation, simplicity and executable nature of PNs model make PNs suitable for simulation, rapid prototyping and verification of systems [21]. Most graph based languages have their root in Petri Net theory, which was applied in workflow modeling for the first time in 1977 [6]. According to [24], PN is a directed graph that mainly consists of two different nodes, places and transitions, Places represent possible states of the system, Transitions are events or actions which cause the change of state [13] [26]. However, attempts to use PNs in practice revealed two serious drawbacks. First of all, there were no data concepts and hence the models often became excessively large, because all data manipulation had to be represented directly in the net structure. Secondly, there were no hierarchy concepts, and thus it is not possible to build a large model via a set of separate sub-models with well-defined interfaces.

High Level PNs (HPN) and Low Level PNs (LPN) [27] are types of PN. HPN supports abstract data type and state transition with data processing but LPN does not have data type and data processing mechanism. LPN and HPN are chosen depending on what kind of system to be modeled. Generally, analysis of LPN is comparatively easy, but a net generally grows large. HPN can express a system in a compact net, but, on the other hand, analysis of HPN is difficult. Colored Petri Net (CPN) [24] incorporates both data structuring and hierarchical decomposition without compromising the qualities of the original Petri Net and thus removed these two serious problems of PN. Timed PNs [28] introduced time in Petri nets, and Hybrid PNs [29] can model a system where discrete state transitions and continuous state transitions coexist. PN theory and applications are discussed in detail in [30]. Flow Charts, Data Flow Diagrams, Role Activity Diagrams, Role Interaction Diagrams, and Integrated Definition for Function Modeling are also approached for graph based and it discussed in details in [24].

Object Oriented Petri Net Modeling (OOPN) is defined on a collection of elements comprising constants, variables, net elements (places and transitions), class elements (object nets, method nets, synchronous ports, and message selectors), classes, object identifiers, and method net instance identifiers. OOPN is applied in different domains for examples [25] Technical Computer Science (modeling & simulation of distributed and concurrent systems, modeling of network protocols, OO operating systems, real time and embedded systems), Software Engineering (modeling of graphical user interfaces, development of information systems, design of database applications, and prototyping of OO design models, and Information Systems (enterprise modeling, BPM and analysis, office information systems, workflow systems, logistic control, production control, flexible manufacturing systems, and automation technique.

III. CHANGE MANAGEMENT ISSUES IN BPM

BPM has been successfully introduced and implemented in several application fields such as Healthcare, Higher Education, Effective Customer Relationship Management, and Customizable Product Manufacturing. Therefore, its significance has increased dramatically. Current varying market opportunities are commented as “Change has become the only certainty” [23]. To stay efficient and effective in such a turbulent environment, organizations are required to adapt their structures and business process to new conditions continuously [31]. As a response, organizations should provide new technologies to manage their dynamic changes in business process.

Many companies use BPMS for modeling and execution support of their business process. Many processes are highly dynamic and require changes even during execution. Common commercial BPMS fail to support such processes appropriately since they work in a rather static manner; they demand that the structure of a process is fixed before execution [8]. Also according to [32] most commercial BPMS support rapid adaptations of workflow definitions but prohibit dynamic changes in workflow instances, like adding missing activities. Thus, they cannot optimally support dynamic processes. This problem gave rise to a new research field. The aim is to design and control the organizational structures in a very flexible way so they can rapidly adapt to changing environments [1]. Enterprise requirements highlight flexible and adaptive processes whose execution can evolve according to situations that cannot always be prescribed, and/or according to business changes (organizational, process improvement, strategic). In order to achieve flexible business process and make quick response to varying requirements, it is extremely important to propagate the change in progress instances [8].

Software systems are not static, and so the business process management technology continues to face challenges in coping with dynamic business environments [2]. Software systems must change and adapt to the environment, or become progressively less useful. Increasingly, today’s software engineers need systematic and methodical approaches for change analysis and management

IV. RELATED WORK

Flexible Modeling and Execution of Workflow Activities model suggested by [33] is based on the activity meta-model. The system supports the functionalities of flexibility, and dynamic changes such as add and delete activities but no activity can be in state running during the change time.

ADEPTflex [34] is a graph based workflow model to support dynamic changes of running workflow instances and concentrate on structural changes and support the users to modifying the structure of a running workflow, while maintaining its structural correctness and consistency.

A special kind of dynamic change called compatible change based on Petri Net proposed in [35]. An algorithm is put forward to calculate the minimal region affected by the changes. Furthermore, it proves that the change regions can be used to check the compatibility of workflow changes (change can be applied to the workflow process, without causing any structural errors or behavioral inconsistencies). It is applicable and efficient in terms of time and space for large-scale and complex systems.

Constraint-Based Flexible business process management approach [2] based on the notion of process constraints was developed to demonstrate how the specification of selection and scheduling constraints can lead to increased flexibility in process execution, while maintaining a desired level of control. A key feature and strength of the approach is to use the power of constraints, while still preserving the intuition and visual appeal of graphical languages for process modeling. Also a number of aspects addressed in this approach including constraint modeling, specification and validation, as well as run time functions for supporting the evolution of business process.

Simulate Dynamics on a Static BPMS approach [32] extends the static BPMS in as much as dynamic changes of processes during execution. The approach described how support for run time dynamics, e.g. dynamic modifications of workflows (Add, Remove and Iteration), can be realized by an additional dynamics layer based on existing static BPMS. The approach used the Web Services Business Process Execution Language (WS-BPEL) construct types at the run time to modify the static BPMS. This approach used a small subset of all WS-BPEL construct types and deal with problems arising from the concurrency and distribution of workflows; this is the main limitation of this approach.

According to [36] business process management would greatly benefit from integration with business rule management. But there is still no established solution to this integration problem, and the leading BPML, BPMN, does not provide any explicit support for rules. Combination between rules and Activities for modeling Service-Based business process system is proposed to cover this problem. The approach investigates the extension of BPMN by adding rules (R2ML) as a modeling concept in the form of a new gateway type, using the principles of Model-Driven Engineering. The integration on the level of the meta-models of the involved languages, a new rule-based process modeling language called rBPMN (Rule-based BPMN) can be changed during the runtime.

Nested Nets for Adaptive Systems proposed in [37] to model adaptive workflow in health care environment. Nested nets are PNs in which tokens can be PNs themselves. This means that processes are considered as objects that can be manipulated by other processes construct more flexible workflow management systems that can be modified during the execution. Integration of OO Design with CPNs approach was developed by [21] [38] to check the correctness of the designed system. The approach integrates OO techniques at the design level and use of CPNs at the verification and validation level. OO methodologies lack analysis and verification methods and PNs are suitable for validation and verification of systems. The approach presents a technique to transform an OO design into hierarchical CPNs model with Abstract Node approach. Abstract Node is a unified abstraction construct of Petri Nets. Abstract Node can be both abstract place to hold data and abstract transition to process data. An example of ATM is presented to illustrate the Abstract Node method.

Object Oriented Petri Nets with modularity (OOMPNets) model [39] is an advanced CPN that introduces CPNs into OO techniques. Based on the scenario, OOMPNets can be allowed to describe objects incrementally. The analysis techniques based on CPNs can be applied for that of OOMPNets to reduce the effects of specification errors. OOMPNets supports gradual progress on modeling software requirement with formal representation from actor, data views, control flow and data flow. The incomplete specifications are encapsulated in nodes with hierarchical presentation to support forward and backward traces. The flexibility for presenting incomplete specifications in a formal format can allow the analysis for these specifications by those techniques in CPN.

Reflective PN approach was developed by [7] [40] to model a (discrete-event) system and separately all its possible evolutions, and to dynamically adapt the system's model when evolution must occur. The approach is structured into two logical layers, The first layer called base-level is represented by the PN modeling, whereas the second layer called meta-level. In the meta-level, a CPN composed by the evolutionary strategies that will drive the evolution of the base-level PN when certain events occur. Entities on the meta-level perform computations on entities residing on the lower level.

Reconfigurable Object Nets (RONs) [41] are the integration of transition firing and rule-based net structure transformation of place/transition nets during system simulation. RONs are high-level nets with two types of tokens: object nets (place/transition nets) and net transformation rules (a dedicated type of graph transformation rules). Firing of high-level transitions may involve firing of object net transitions, transporting object net tokens through the high-level net, and applying net transformation rules to object nets. Net transformations include net modifications such as merging or splitting of object nets, and net refinement. This approach increases the expressiveness of PNs and is especially suited to model mobile distributed processes. Approaches in [42] [43] are an examples of runtime updates based on Java programming, where [42] focus specifically on dynamic Java (focus on being full consistency with the Java language and its type safety). They extended the default Java class loader in a way that class definitions can be replaced and objects or dependent classes can be updated. The replacement is initiated by the user by explicit calls to the class loader in the application program. In [43], interface changes are allowed and do not require the application to be developed with evolution in mind based on dynamic delegation with Lava (a variation of Java).

V. Proposed Framework

The proposed Object Oriented Coloured Petri net (OOCPN) framework for modeling and analysis of change management in BPM is a High level PN that supports OO concepts. It aims to integrate the UML modeling diagrams and CPN modeling to support the dynamic changes of process during runtime. It supports corrective and evolutionary changes in the process modeling from the control flow perspectives [20]. The framework is divided into two phases: design time and runtime development as shown in Fig. 1.

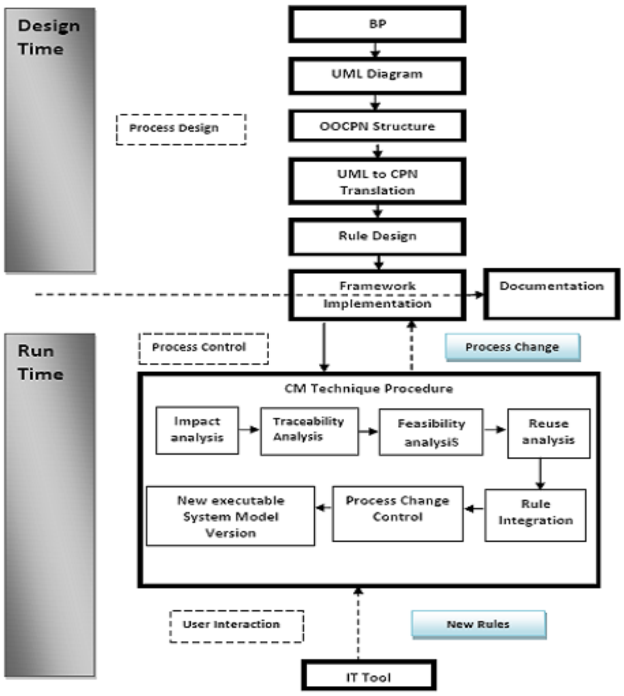

Fig. 1: Change Management in BPM Framework

A. Design Time Development

Fig. 2 presents the stages of the design time development in the framework.

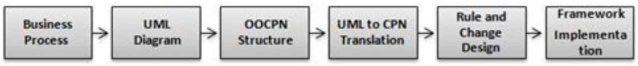

Fig. 2: Design Time Development

UML Model Design: The business process model defined is based on an OO modeling language (UML). UML model supports the object characteristics and relationships between various objects including abstraction, inheritance, polymorphism and dynamic binding. UML modeling tool like Rational Rose tool can be used for designing and partially checking the correctness of the UML diagram.

OOCPN Structure: OOCPN structure in this framework is a mutual integration of OO techniques and CPNs. It is perceived as a further development of embedding PN models into objects. CPNs are thus used to model both the inner behavior of objects, and the inter-object communications. Objects will be initially used to determine the structure of a system. Subsequently, the behavior of the objects is modeled with the help of PNs [44]. It supports dynamic of object creation, use relationship, and inheritance. Two OOPN structures are proposed to implement this framework: OOCPN and OOCPN'. The OOCPN structure includes the CPN elements which represent the system model after translating it from the UML model. OOCPN' structure is to implement the runtime rule (change). It will be composed of OOCPN identifications, and change management identification to identify the notations for change management elements. Two structures are used to support the change management techniques as discussed in the rule integration in runtime development Section B. The framework assigns identifiers automatically when a new version is submitted, and it stores the differences between the two structures (i.e. OOCPN structure stores the system model and OOCPN' structure stores the new rules model only). The formal definitions of these structures are as follows:

Definition 1. OOCPN structure [45]: OOCPN = ($\sum$, P, T, A, N, C, G, E, M_0, I, O) where:

$\sum$: is a finite set of non-empty types, called colour sets
P: Finite set of places
T: Finite set of transitions
A: represents a set of directed arcs
N: is a node function
C: is a colour function
G: is a guard function
E: is an arc expression function
M_0: is the initial (colored) marking
I: is a function which determines the input multiplicity for each input arc
O: is a function which determines the output multiplicity for each output arc

Definition 2. Change Management Identification (CMI):
CMI = (LC, GC, S) where:
LC: Local Changes, representing the internal object change.
GC: Global changes, representing the object communication change include add/delete objects.
S: Status, used to determine the status of the process, i.e. whether it is active or not. It is used in the versions of the updated model; the values of S are used to check the status of the new rule as shown in Algorithm 1 and Fig 1.

If New Rule is Create Process then
    S is active for this New Rule
Else if New Rule is Modify Process then
S is active for this New Rule and inactive for the old process
Else if New Rule is Delete Process then
    If S is not active then
        No change in S value.
Else Wait until the process is closed, and then the value of S will be not active.
Else "this type of change is not supported by the framework"

Algorithm 1: New Rule Status

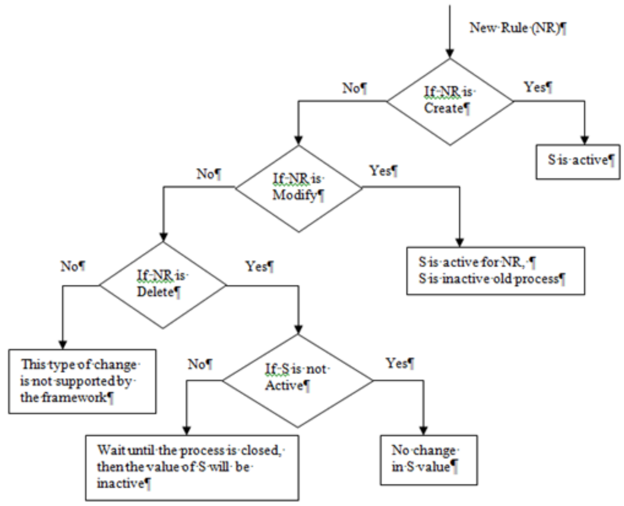

Fig 3. New Rule Status

Definition 3 OOCPN' structure: OOCPN' is a combination between OOCPN structure and CMI. The relation between these structures and how it will be used for change management is discussed in the rule integration in runtime development in Section B.

UML to CPN Translation: UML diagram will be translated to CPN based on the suggested OOCPN structure. The translation is based on CPN tool; it loads the UML diagram, and then, translates it to CPN format automatically according to the suggested OOCPN structure. The tool will provide the property of verifying the correctness of the CPN model.

Rule Design: The rule design includes reaction rules that can be added into the process models. The important point is a place where rules should be used, and this is based on PN modeling language. According to [46] the rule is a combination of 4 parts: Condition (Expression on the arc in PN), Action (Expression on the arc in PN), Rule (Transitions in PN), and Fact (Reachable variables in PN). The rule implements the change in the business process at the runtime (represents the dynamic behavior). The design of the rule will be on two stages:

At the design time, A GUI will be designed to enter the rule at the runtime, and the design of the rule will be based on the OOCPN' structure especially the runtime change management parameters (LC, LG, and S).

At the runtime, the rule will be entered through the GUI in a PN format, the framework will verify the correctness of the rule, after that, the new rule will be integrated with the system model and it will be part of it according to the procedure in Section B.

B. Runtime Development

The types of changes the framework support are the Corrective and Evolutionary changes in the system, where the corrective changes aim to correct a design error on the process definition or to react to an exception which happens during the execution of an instance, and the evolutionary is required due to the redesign or reconfiguration of the business process. Changes in a business process model will be the result of: Create Process (Adding some activities to the business process model, this includes creating a new Place or Transition) , Delete Process (Removing some activities from the business process model, this includes deleting a Place or Transition), or Modify Process (Modifying activities in the business process model, this includes modifying a Place or Transition). Fig. 4 presents the stages of the run time development in the framework,

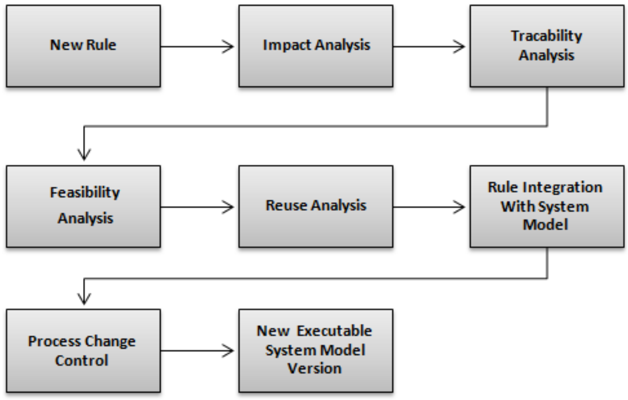

Fig 4. Runtime Change Management Stages

The stages will be according to the following procedure:

1) Entering a new rule through the GUI. The rule will be part of OOCPN' structure and according to its format. The correctness of the rule will be checked by the CPN tool.
2) Impact analysis: The ability to reason about the potential effects of change on software process (local or global impact), where LC implies Local change and GC implies the global change.
3) Traceability analysis: An analysis of the dependencies between and across software artifacts and actors at all levels of the software process. Traceability analysis is important in change impact [47].
4) Checking the feasibility of the change: it will be checked based on the rule design in the design time development, also according to the impact analysis and type of changes the framework will support, and this will be determined by the change management parameters (LC, LG, and S).
5) Reuse analysis.
6) Integration the rule with the system model: The new rule will be entered through the GUI, then it will be stored in the OOCPN' structure after checking its correctness by CPN tool. It will be stored in the same format and elements as in OOCPN combined with the CMI elements. Updating the system model (updating means create, delete, or modify process) will be according to algorithm 2. Then, the new rule will be integrated with the system model (OOCPN structure) and it will be part of it. The correctness of the integrated part and the synchronization problems such as deadlock will be checked automatically based on the CPN tool.

If New Rule is Create Process then

- The new model is the union between the new rule with the OOCPN structure.
- Running the Algorithm 1 for updating the value of S.

- Message will be sent to inform that new updates are available

Else If New Rule is Delete Process then

If S for this process is not active then

- Checking the integrity and inconsistency of the system (leaving the system in a consistent state).
- The process will be deleted permanently.

Else

- Algorithm 1 will run to update the values of S.
- Checking the integrity and inconsistency of the system (leaving the system in a consistent state).
- The process will be deleted permanently.

Else if New Rule is Modify Then

- Algorithm 1 will run to update the values of S (Active for the new rule and inactive for the old rule).
- Checking the integrity and inconsistency of the system (leaving the system in a consistent state).
- If the process is running, then a message will be sent to inform that new updates are available and wait until the process is closed.
- If the process is closed then the new model is the union between the new rule and the OOCPN structure. The old process will not be deleted from the system model; it will be part of it until it is deleted by Delete Process.

Else "The new rule is not supported by the framework".

Algorithm 2. New Rule Integration

7) Process change control: When performing configuration changes it is important that information is not lost and that we leave the application in a consistent state. This includes checking the integrity and consistency of the change using CPN tools (Verification and Validation).
8) New executable system model version to simulate the change management.

C. Outputs and Results

The excepted outcomes for this framework are:

Short Term Outcomes: a framework to simulate the following:

- Translating the UML model to CPN according to the suggested OOCPN structure.
- Integrating new rule with the system model.
- Applying new change management technique for dynamic BPM changes in runtime processes.
- Long Term Outcomes:
- Effective change management technique for the dynamic BPM framework based on the combination of UML diagramming technique and CPN modeling languages.
- Increasing the representation capability for BPM to support runtime environment criteria of flexibility, adaptability and dynamic.
- Applying the system in a higher education environment and increases the flexibility and adaptability of the processes.

D. Framework Analysis and Evaluation

The important characteristics in the analysis of the models are: estimation of complexity, process size, effort of testing, effort of maintenance, understandability and quality to enable the correction of problems before any drastic consequences occur, and testing involving human subjects and their perception of process complexity. Also according to [48], completeness, correctness, and change realization are the three fundamental issues that should be considered as regards to dynamic changes in business process. According to [49], methods of analysis of PN may be classified into the following three groups:

1) Coverability (Reachability) tree method.
2) Matrix equation approach.
3) Reduction or Decomposition techniques.

The first method involves essentially the enumeration of all reachable markings. It should be able to apply to all classes of nets, but is limited to small nets due to the complexity of the state-space explosion. On the other hand, matrix equations and reduction techniques are powerful but in many cases they are applicable only to special subclasses of PNs or special situations

In this framework, CPN analysis techniques are suggested to analyze the framework including correctness of the source analysis and results as well as statistics data provided by the CPN analysis tool, checking the consistency, reliability, verification and validation. Validation can be done by interactive simulation provided by CPN tools, for verification and performance analysis more advanced analysis techniques are needed. Fortunately, many powerful analysis techniques have been developed for PN [49]. Linear algebraic techniques can be used to verify many properties, e.g. place invariants, transition invariants, and (non-)reachability. Coverability graph analysis, model checking, and reduction techniques can be used to analyze the dynamic behavior of a Petri net.

VI. Summary

Runtime change management in BPM framework is to provide an adaptive, flexible, and dynamic framework for process management. It helps the software engineers by finding a systematic and methodical approach for change analysis and management, also for modeling and analyzing the runtime changes in BPM. The framework is a high-level PN that supports OO concepts. It integrates the modeling design features of UML with the powerful instrument for modeling, analyzing, and simulating dynamic systems with concurrent and non-deterministic behavior of the CPN. The main features of this framework provide the following:

- New OOCPN structure, by adding parameters related to the change management, These parameters are used in the runtime to check the feasibility of the change and determine the type of the change that the system supports and its impact analysis. The techniques of translating the UML model to CPN model is based on the new OOCPN and CMI structures proposed in this framework.

- Two new algorithms for integrating a new rule (change) with the system model.
- New change management procedure for the runtime changes as described in runtime development in Section B.
- It supports the corrective and evolutionary changes in the process modeling from the control flow perspectives. And it can be applied in any application field for supporting these two types of changes. The framework will check the availability and feasibility of the change.

REFERENCES

[1] S. Nurcan, "A Survey on the Flexibility Requirements Related to Business Processes and Modeling Artifacts," in Proceedings of the 41st Hawaii International Conference on System Sciences HICSS'2008, 2008, pp. 378-378.

[2] R. Lu, "Constraint-Based Flexible Business Process Management," in School of Information Technology and Electrical Engineering, University of Queensland, 2008.

[3] The Forrester consulting, "Software Challenges of Software Change Management in Today's Siloed IT Organizations," A Commissioned Study 2006.

[4] D. Koomsub, "A Case Study of Change Management of ERP Implementation Project Using SAP R/3," Thailand, Independent Study Project 1999.

[5] S. Sadiq, W. Sadiq, and M. Orlowska, "A Framework for Constraint Specification and Validation in Flexible Workflows," , 2005.

[6] R. Lu and S. Sadiq, "A Survey of Comparative Business Process Modeling Approaches," in International Conference on Business Information Systems (BIS 2007), 2007, pp. 82-94.

[7] L. Capra and W. Cazzola, "A Reflective PN-based Approach to DynamicWorkflow Change," in Ninth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, 2007, pp. 533-540.

[8] D. Zhang, D. Cao, L. Wen, and J. Wang, "An Efficient Approach for Supporting Dynamic Evolutionary Change of Adaptive Workflow," in APWeb 2008, 2008, pp. 684-695.

[9] C. Lakos, "Object Oriented Modelling with Object Petri Nets," in Concurrent object-oriented programming and petri nets: advances in petri nets.: Springer-Verlag New York, Inc. Secaucus, NJ, USA , 2001, pp. 1 - 37.

[10] M. zur Muehlen, M. Indulska, and G. Kamp, "Business Process and Business Rule Modeling: A Representational Analysis," in Eleventh International IEEE EDOC Conference Workshop, 2007, pp. 189-196.

[11] B. Von Halle, Business Rules Applied: Building Better Systems using the Business Rules Approach.: Wiley, 2001.

[12] O. Thomas, T. Dollmann, and P. Loos, "Towards Enhanced Business Process Models Based on Fuzzy Attributes and Rules," in Proceedings of the 13th Americas Conference on Information (AIS 2007), Keystone, Colorado, USA. Atlanta, Georgia, USA, 2007.

[13] A.-W. Scheer, "ARIS toolset: a software product is born," Information Systems, vol. Vol. 19 No. 9, pp. 607-24, 1994.

[14] G. Knolmayer, R. Endl, and M. Pfahrer, "Modeling Processes and Workflows by Business Rules. In Business Process Management," LCNS 1806, Springer-Verlag Berlin Heidelberg, pp. 16 - 29, 2000.

[15] J. Mendling, M. La Rosa, and A. H. ter Hofstede, "Correctness of Business Process Models with Roles and Objects," QUT ePrints Technical Report #13172 2008.

[16] L. Zeng, D. Flaxer, and H. J. Chang, "Jun-Jang PLMflow: Dynamic Business Process Composition and Execution by Rule Inference ," in In proceedings of 3rd VLDB Workshop on Technologies for E-Services (TES'02), Hong Kong, China, 2002.

[17] N. F. P. Jennings, T. Norman, P. O'Brien, B. Odgers, and J. Alty, "Implementing a Business Process Management System using ADEPT: a Real-World Case Study," International Journal of Applied Artificial Intelligence, vol. Vol 14, no. 5, pp. 421-463, 2000.

[18] M. Pesic and W. van der Aalst, "A Declarative Approach for Flexible Business Processes Management," in Business Process Management Workshops, 2006, pp. 169-180.

[19] Object Management Group. (2004) Business Process Definition Metamodel. Version 1.0.2. [Online]. HYPERLINK "http://www.bpmn.org/Documents/BPDM/OMG-BPD-2004-01-12-Revision.pdf" http://www.bpmn.org/Documents/BPDM/OMG-BPD-2004-01-12-Revision.pdf

[20] N. Russel, W. M. van der Aalst, A. H. ter Hofstede, and P. Wohed, "On the Suitability of UML 2.0 Activity Diagrams for Business Process Modeling," in Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling, vol. 53, 2006, pp. 95-104.

[21] E. B. Bauskar and B. Mikolajczak, "Abstract Node Method for Integration of Object Oriented Design with Colored Petri Nets," in Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06), 2006, pp. 680 - 687.

[22] M. Owen and J. Raj. (2003) BPMN and Business Process Management : Introduction to the New Business Process Modeling Standard. [Online]. HYPERLINK "Popkin%20Software%20%20www.popkin.com" Popkin Software www.popkin.com

[23] H. Smith and F. Peter, "Buseniss Process Modeling third wave," 2003.

[24] R. S. Aguilar-Sav'en, "Business process modelling: Review and framework," International Journal of Production Economics, vol. Vol 90, no. 2, pp. 129-149, 2004.

[25] M. Zapf and A. Heinzl, "Techniques for Integrating Petri-Nets and Object- Oriented Concepts," 1999.

[26] J. Tick, "P-Graph-based Workflow Modelling" Acta Politechnica Hungarica," , vol. ISSN 1785-8860, Vol. 4, 2007, pp. 75-88.

[27] T. Miyamoto and S. Kumagai, "Application of Object-Oriented Petri Nets to Industrial Electronics," in The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON), Taipei, Taiwan, 2007, pp. 64-69.

[28] M. Holliday and M. Vernon, "A Generalized Timed Petri Net Model for Performance Analysis," , vol. 13 (12), 1987, pp. 1297-310.

[29] J. LeBail, H. Alla, and R. David, "Hybrid Petri nets.," in In Proceedings 1st European Control Conference, Grenoble, France., 1991.

[30] V. Kordic, Petri Net, Theory and Applications. Vienna, Austria : I-Tech Education and Publishing, 2008.

[31] W. van der Aalst, A. ter Hofstede, and M. Weske, "Business Process Management: A Survey," in Proceedings of Conference on Business Process Management (BPM 2003), Eindhoven, Netherlands, 2003.

[32] R. Wo"rzberger, N. Ehses, and T. Heer, "Adding Support for Dynamics Patterns to Static Business Process Management Systems," in Proceedings of the 7th International Symposium on Software Composition (SC 2008), LNCS, vol. 4954, 2008, pp. 84-91.

[33] M. Weske, "Flexible modeling and execution of workflow activities," in Proceedings of the Thirty-First Hawaii International Conference , vol. 7, Jan. 6-9, 1988, pp. 713-722.

[34] M. Reichert and P. Dadam, "ADEPTflex - Supporting Dynamic Changes of Workflows without Losing Control," Journal of Intelligent Information Systems, Special Issue on Workflow Management, Vol. 10, pp. 93-129, 1998.

[35] P. Sun and C. Jiang, "Analysis of workflow dynamic changes based on Petri net," in Information and Software Technology, 2008.

[36] M. Milanović, D. Gašević, and G. Wagner, "Combining Rules and Activities for Modeling Service-Based Business Processes," Munich, Germany, 2008.

[37] K. M. van Hee et al., "Nested nets for adaptive systems," in Lecture Notes in Computer Science : Petri Nets and Other Models of Concurrency - ICATPN 2006, vol. 4024, 2006, pp. 241-260.

[38] H. Motameni, A. Movaghar, B. Shiraz, B. Aminzadeh, and H. Samadi, "Analysis Software with an Object-Oriented Petri Net Model," World Applied Sciences Journal, vol. 3, no. 4, pp. 565-576, 2008.

[39] C. H. Wang and F. J. Wang, "An Object-Oriented Modular Petri Nets for Modeling Service Oriented Applications," in 31st Annual International Computer Software and Applications Conference(COMPSAC 2007), vol. 2, 2007, pp. 479-486.

[40] L. Capra and W. Cazzola, "A Petri-Net Based Reflective Framework," in Procedings of of the IPM International Workshop on Foundations of Software Engineering (FSEN'05), Tehran, Iran, 2005, p. 41–59.

[41] E. Biermann, C. Ermel, F. Hermann, and T. Modica, "A Visual Editor for Reconfigurable Object Nets based on the ECLIPSE Graphical Editor Framework," in Procedings of 14th Workshop on Algorithms and Tools for Petri Nets (AWPN'07), Universit ät Koblenz-Landau, Germany, 2007.

[42] S. Malabarba, R. Pandey, J. B. Gragg, and J. F. Barnes, "Runtime support for type-safe dynamic java classes," in Proceedings of the 14th European Conference on Object-Oriented Programming, 2000.

[43] P. Costanza, "Dynamic object replacement and implementation-only classes," in In Proceedings of the 6th International Workshop on Component-Oriented Programming at ECOOP, Budapest, Hungary, 2001.

[44] R. Bastide, "Approaches in unifying Petri Nets and the Object-Oriented Approach," in Proceedings of the 1st Workshop on Object-Oriented

Programming and Models of Concurrency within the 16th International Conference on Application and Theory of Petri Nets, Turen, 1995.
[45] V. Kordic, Petri Net, Theory and Applications. Vienna, Austria, Vienna, Austria : I-Tech Education and Publishing, 2008.
[46] C. Tadj and T. Laroussi, "Dynamic Verification of an Object-Rule Knowledge Base Using Colored Petri Nets," Journal of Systemics, Cybernetics and Informatics, vol. Vol 4 , No.3, 2006.
[47] W. Lam, V. Shankararaman, S. Jones, J. Hewitt, and C. Britton, "Change analysis and management: a process model and its applicationwithin a commercial setting," in IEEE Workshop on Application Specific Software Engineering Technology Proceedings, 1998, pp. 34 - 39.
[48] S. Rinderle, M. Reichert, and P. Dadam, "Correctness Criteria for Dynamic Changes in Workflow Systems: A Survey," Data & Knowledge Engineering, vol. 50, no. 1, pp. 9-34, 2004.
[49] T. Murata, "Petri Nets: Properties, Analysis and Applications," Proceedings of the IEEE, vol. 77, no. 4, pp. 541-580, 1989.
[50] B. Rajabi and S. P. Lee, "Runtime Change Management Based on Object Oriented Petri Net," in International Conference on Information Management and Engineering (ICIME 2009), Kuala Lumpur, Malaysia, 2009, pp. 42-46.
[51] B. Rajabi and S. P. Lee, "Change Management in Business Process Modeling Survey," in International Conference on Information Management and Engineering (ICIME 2009), Kuala Lumpur,Malaysia, 2009, pp. 37-41.
[52] B. Rajabi and S. P. Lee, "Change Management in Business Process Modeling Based on Object Oriented Petri Net," in Proceedings of World Academy of Science, Engineering and Technology, vol. Volume 38, Penang, Malaysia, 2009, pp. 12-17.
[53] B. Rajabi and S. P. Lee, "Change Management in Business Process Modeling Based on Object Oriented Petri Net," International Journal of Business, Economics, Finance and Management Sciences, vol. Volume 1 Number 1 2009, pp. 72-77, 2009.

Bassam Atieh Rajabi received his B.S. degree in Computer System Engineering from Palestine Polytechnic University, Hebron, Palestine, in 2001 and the M.Sc. degree in Computer Science from Alquds University, Jerusalem, Palestine, in 2005. Currently, he is a PhD student in Computer Science at University Malaya, Malaysia. From 2001 to 2004, he was a Research and Teaching Assistant with the Computer Science Department, Alquds University, Jerusalem, Palestine. From 2001 to 2005 he was a Lecturer with the Computer Science Department, ORT College, Jerusalem, Palestine. He was a Lecturer and Dean Assistant for Administrative Affairs From 2005 to 2008 with Wajdi Institute of Technology, Jerusalem, Palestine. His areas of interest are Software Design and Modeling Techniques.

Sai Peck Lee is a professor at Faculty of Computer Science & Information Technology, University of Malaya. She obtained her Master of Computer Science from University of Malaya, her Diplôme d'Études Approfondies (D. E. A.) in Computer Science from University of Pierre et Marie Curie (Paris VI) and her Ph.D. degree in Computer Science from University of Panthéon-Sorbonne (Paris I). Her current research interests include Software Reuse, Application and Persistence Frameworks, Requirements and Design Engineering, Object-Oriented Techniques and CASE tools. She has published more than 80 research papers in local and international journals and conferences.