From BPMN to live application: How the context can drive an auto-adapted system

The Can Do
Université Côte d'Azur
CNRS, I3S
Nice, France
The-Can.DO@etu.unice.fr

Gaëtan Rey
Université Côte d'Azur
CNRS, I3S
Nice, France
Gaetan.REY@unice.fr

Jean-Yves Tigli
Université Côte d'Azur
CNRS, I3S
Nice, France
Jean-Yves.TIGLI@unice.fr

Stéphane Lavirotte
Université Côte d'Azur
CNRS, I3S
Nice, France
Stephane.LAVIROTTE@unice.fr

Nhan Le Thanh
Université Côte d'Azur
CNRS, I3S
Nice, France
Nhan.LE-THANH@unice.fr

***Abstract*—Today, with requirements about improving the adaptation ability and reducing the complexity on building application of Auto-adapted system, it will be very important to use contextual concerns such as BPMN, CTT, Scenario, etc. as a tool to model all the relevant concerns of context. In this paper, we provide a new architecture that using Context-aware management to manage Auto-adapted system and using the Context intermediate model to redefine useful data exported from the BPMN tool at design time. Our aim is providing a solution to use specific views in design time to make simpler expert's task. Moreover, it also opens an ability to use many independent views on AAS as the way to separate concerns and improve responsiveness ability in a dynamic context-aware system.**

Keywords—Context-aware, Ubiquitous computing, Context modeling, Middleware.

I. Introduction

In the last decade, the introduction about Ubiquitous computing of Weiser [1] and context definition of Dey [2] opened many research directions about an Auto adaptive system, context-aware, context modeling and so on. Many researchers have done on the adaptation domain [3] or modeling context at design time [4], [5] to improve system adaptation ability and capacity of the designer to model all relevant concerns of context [6]. Their research has created many architecture frameworks [7], [8] and multiple meta-models of context [9]. However, their results also make new problems such as increase complication of expert's task [10], take more time to adapt properly, the risk of conflict between multiple applications [11].

With an auto-adapted system, if we can use specific contextual concerns such as Business process modeling notation (BPMN), ConcurTaskTrees (CTT), Scenario, etc. at design time to model context, we can solve the challenges of the system such as:

- Simpler expert's task.

- Increase the ability to reuse of views.

- Separate concerns and responsiveness of the auto-adapted system.

When we use each contextual concern as a specific view to model context, we must find the solution to manage and execute data from specific views. In our previous work [12] we propose to use Context-aware management (CAM) to manage specific views, it is the solution to manage context elements and improve the adaptation ability of a dynamic context-aware system. In this paper, we describe details about the way how to use an independent view as BPMN to drive an auto-adapted system, and we introduce the notion of a Valid and Invalid state which necessary for a real operation of the auto-adapted system on manufacturing.

In the manufacturing domain, we have many BPMN tools which can be used to describe the graphical document, diagram and simulate processes in industry-standard BPMN. However, in our Auto-adapted system (AAS), the data exported from a BPMN tool cannot be executed directly by CAM systems. Moreover, the exported data from BPMN still has somewhat unsatisfactory as a schema formalism. It stores too much information not necessary for CAM and missing some useful things such as structural, validation constraints and execution order of the tasks. It is motivation to us propose using Context intermediate model (CIM) to convert BPMN data into the format of CAM standard. This paper also identifies which elements BPMN lacks in terms of CAM standard and suggests a solution to solve conflicts between adaptation rules at run-time.

The remaining sections of the paper are organized as follows: Section 2 presents related works on a context-aware application and modeling context use the specific view. In section 3, we discuss the Model architecture of the AAS. Section 4 presents the XPDL translation process use Context intermediate model. Then we describe the implementation of our approach in section 5. Section 6 discusses limited of approach and propose a solution to solve. Finally, section 7 offers our conclusions and future work.

II. Related Work and Motivation

We can find a lot of work have been done in the context-aware applications in the past few years. In this section, we present a selection of works that focus on building a context-aware application based on models and using specific views on modeling context to improve separation concerns and adaptation capacities of AAS.

- Dey et al. [13] develop a context toolkit based on composed of the sensor to build context-aware system. Their study supports a conceptual framework that separates the acquisition and representation of context from the delivery and reaction to context by a context-aware application. The context toolkit provides methods to access to such context information, transform the context information into high-level formats that are easier to handle for AAS. They also demonstrate how such a framework can support the investigation of important research challenges in the area of context-aware computing. However, these systems do not provide a solution to make simpler expert's task. This approach still has some limitation in the schema formalism of context information, so it is becoming complicated to use in a large self-adaptive system [14] with the wide range of context which can combine many views in an application.

- CASAS [5] presents a new model architecture and platforms which is based on the usage of both global and local contexts by two independent modeling techniques in design time. This approach used a semantic data model to describe the global context and the local contexts are derived as views depend on each application. The main contribution of this approach can be much more efficient due to smaller, simpler and better tailored local context models. It also allows developers to use high-level modeling techniques in design time. This approach provides a solution to help context information can be shared between different applications, increasing reuse of context information and reducing their complexity. However, this approach only focuses on mapping between local and global context, they need to extend their work with complex event processing in order better support and improve the efficiency of reasoning and run-time adaptations [15].

- The work has been done by Boris Chidlovskii [16] proposes the XML query algebra based the tree automata model. This model uses tree automata as the schema for XML files. They defined tree automata provided ability to design the XML query language in the way similar to the relational algebra and induce precise schema for any XML query formulated in this language. They addressed the problem of tight XML schema in modeling XML documents and introduce a novel mechanism based on tree automaton. This approach opens the new way to use special view such as ConcurTaskTrees, BPMN, etc. in a modeling context. However, it proposes only guidelines to support the translation of tree automata into DTD without a solution to separate context concerns and combine special many contextual concerns into an application of AAS.

- CAISDA [17] requires designer must build its application context by using modeling tools. This approach focuses on a software framework that observing the execution of an application. When the system detects the request of the user in run-time, the AAS in CAISDA will analyze and update all changes of context during the system operation execution. The CAISDA supports also a mechanism to help and guide the designer on developing their application. This approach is good for applications which use single view but with applications use many views, the analyzing context will become complicate and take more time with AAS. This study also requires each designer must be expert because they must work on many domains.

- David Schumm et al. [18] show how BPEL processes can be modeled using the graphical aspect of BPMN in order to facilitate modeling of executable processes using BPMN without model transformations. They propose a new solution using graphical representations instead of transformation techniques for creating BPEL processes. They show that it is possible to describe a BPEL process using the icons, connecting elements, and semantics defined in BPMN. The basic principle of this approach is to use BPEL as the meta-model for the graphical process definition. But this solution has limited in the requirement about separate concerns, when we use meta-model [19], [20] that mean the number of adaptation rules was increase and the system take more time to analyze information that may be not related to the current application [21].

- SOCAM [22] is a service-oriented context-aware middleware which supports an architecture for the building and rapid prototyping of context-aware services. They used Ontology Web Language (OWL) to describe a formal context model which address issues including semantic representation, context analyzing and dependency. The SOCAM provides efficient infrastructure support for building context-aware services. It converts various physical space from which contexts are acquired into a semantic space where context can be easily shared a robust system and accessed by context-aware services. This architecture has an intelligent system of reasoning about the context which based on OWL for context modeling. It allows the recording adaptation rules and updates information of context when the system detects context changes. However, this middleware does not provide a solution to separate concerns when the system works in different situations. This approach also makes the work of expert become complicated and system consumption increase.

TABLE 1. COMPARISON ARCHITECTURE OF CURRENT CONTEXT-AWARE FRAMEWORKS.

Context frame work	**Limitation**				
	Separate concerns	Reuse of view	Simpler expert's task	Update context situation	Respons-iveness
Dey et al	+	-	+	-	-
CASAS	-	-	-	+	+
Boris	+	-	-	-	+
CAISDA	-	+	+	-	-
David et al	+	+	-	+	-
SOCAM	+	-	+	-	-

We can see that no single approach has the features to address all limitations of currently context-aware application:

Firstly, with the Context model. Some works [5], [17], [22] use meta-model to represent the static and dynamic aspect of the context. However, in the case of applications that using many contextual concerns, the combination of many specific views in meta-model is very complex and the work of each developer becomes complicated. Moreover, if we want to add, delete or replace specific views that can't be done either by the automatic process. It is motivation to our proposed use CAM to manage views and Context intermediate model which can convert independent views to the standard of CAM. The CAM can add, delete or replace specific views depend on the application. It adapts the requirement about reusable views and improves the responsiveness of the system.

Secondly, with the requirement of a system about Update context situation, Simpler expert task, Separation of concerns. During the adaptation process context can be change and system need Update current context that relates to correctly of adaptations and reduces the decision time of AAS with each application. The works are done in [13], [16] are suitable with some applications that use one specific view. In case combination of specific views, analyzing and selecting context elements at run-time can make the conflict between different applications. Moreover, the work of expert also become complex because they must use knowledge in different domains.

In this article, we present a new architecture for context-aware application using BPMN to model context. Using BPMN view at design time take simplifies the work of designers on building applications of auto adapted system. At run-time, we propose using two independent observation cycling of CAM and AAS as shown in Figure 1. This solution allows CAM focuses on detecting the changing of context to impact on the decision of AAS through Adaptation rules set (AR). At the same time, the AAS focuses only on adaptation to an application based on evaluating the evolution over time of devices.

III. MODEL ARCHITECTURE OF SYSTEM

In our approach, we supposed an architecture as shown in Figure 1 of the Auto-adapted system where we used BPMN as a special view to model context. The Context-aware management can be used to collect information from context and manage the AAS [12].

At design time, we currently introduce a Context intermediate model (CIM) (as shown in Figure 1), between expert specific models BPMN and the Context-aware management. The goal of CIM is transformed specific view into XML description that follows the standard data of CAM. In case of the application uses many special views, each view is described in a totally independent way compared to the other view.

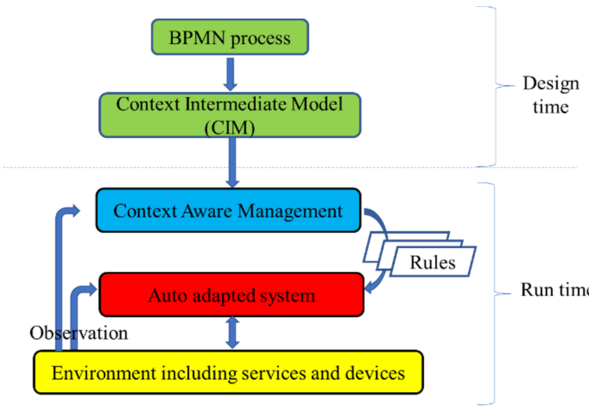

Fig. 1. The special view BPMN on Auto-adapted system.

The Context intermediate model in this approach is an updated version of the Intermediate common model in our previous work [12]. It is an XML description of a Moore automaton described by the following 6-tuple (S, S0, Σ, Λ, δ, λ).

In Figure 2, we show the updated graphics of the XML schema that used to define Context intermediate model in our previous work [12]. In our approach, each special view will be described by a set of states in CIM. Each state defined by a set of predicates. In each state, the designer also associates a list of adaption rules to adapt the application to the current state. In the updated version of CIM, we add one new part of CIM to describe the goals and description of each state that can help to detect the conflict between views at run-time if we have combined more than two different views in an application.

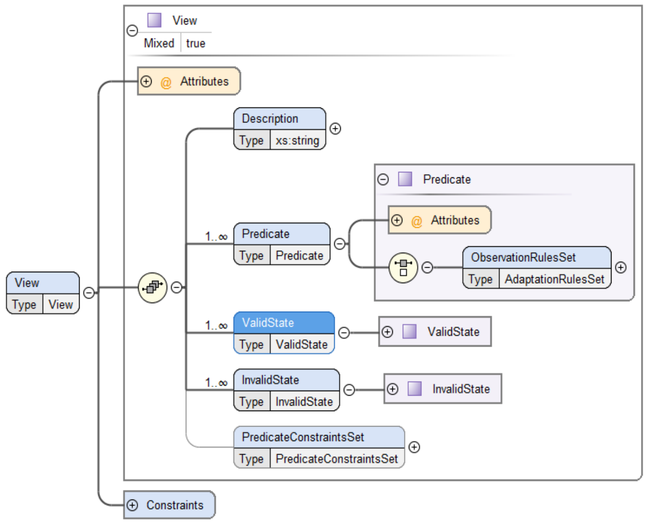

Fig. 2. The Schema of XML program exported from CIM.

To understand about the application of Moore automata theory for Context intermediate model, we can see in below example.

Example 1:

We need to model a scenario of a student who comes laboratory to do some tasks. He must provide an ID card to open the door. When the door opened, the system activates Tool to the student can use for him Task.

We have:

- S_1: ID wrong → $!P_1$ (Predicate 1: Check ID)
- S_2: ID true + Door not opened → $P_1\&!P_2$ (Predicate 2: Open door)
- S_3: ID true + Door opened + Tool not activated → $P_1\&P_2\&!P_3$ (Predicate P_3: Tool activate).
- S_4: ID true + Door opened + Tool activated → $P_1\&P_2\&P_3$

In this Scenario we have 8 Valid States and no Invalid State. Here, we use a special notation to merge some states. For example when we write $P_1\&!P_2$, we describe 2 states define by $P_1\&!P_2\&P_3$ and $P_1\&!P_2\&!P_3$..

- The scenario above can be captured by Moor automaton follow our description in [12], then we have:

$$M = (S, S_1, \Sigma, \Lambda, \delta, \lambda)$$

Where:

- $S = \{S_1, S_2, S_3, S_4\}$
- $\Sigma = \{P_1, P_2, P_3\}$
- $\Lambda \rightarrow$ (observation rules, adaptation rules)
- δ:
$$(S_n(t) * \begin{pmatrix} P1(t) \\ P2(t) \\ P3(t) \end{pmatrix}) = S_n(t+1)$$
(where $P_i(t)$ is the value of predicate P_i at the time t, and $S_n(t)$ is the state S_i selected at the time t)
- $\lambda = \{S_i$, output function$\}$ (i: the number of the set of state

At run-time, we propose using two independent observation cycles of CAM and AAS. This solution can provide more time to the CAM analyze the current context and preserve the responsiveness of AAS. The observation of CAM uses to handle the context information. When the CAM finds a changing in a context such as add, delete, replace special view. It will be required Auto adaptive system reconfiguration. With the independence of the execution cycle of the AAS and CAM, the operations of the application layer are independent of the adaptation process. Thus, the application can continue to operate during the decision stage of CAM and it is interrupted only during the implementation of the adaptation plan when the CAM detects the context switching. That means the AAS can reduce adaptation time and a part of the problem about consumption system. The CAM evaluates each predicate of the view to identify the current situation. Depending on this situation, a list of adaptation rules will be applied and AAS will give suitable action or decision.

It is impossible to solve conflicts between AR at design time on the case of the diversity and unpredictability of developments in a ubiquitous environment. Because it means that we have to predict all concerns that will be associated with the context. Moreover, we must calculate all conflicts that could never take place at execution or real operation of the system. When we analyze the real business processes or security process in a company, it is clear that they should only cover the nominal case. In unexpected situations, we can see that they should not take their qualifier suggests into account because it maybe takes more time of action and decision step in AAS.

IV. REDEFINITION BPMN DATA USE MOORE AUTOMATON IN CONTEXT AWARE SYSTEM

A. Mapping XPDL to CAM standard

The Business Process Modeling Notation (BPMN) is a popular tool which provides a graphical notation for expressing business processes in a Business process diagram [23]. The main purpose of BPMN is providing a process management tool for both business users and technical users.

In our approach, we used a BPMN tool as a special view to model a small part of context that related to using graphical notation in building application. However, the BPMN was only a simple and suitable with business people on daily basis, the execution information from BPMN in CAM is a complicated process. The work of A.White [24] provides a solution to define and execute business processes through XML Process Definition Language (XPDL). It is a standard mechanism to describe an analyzing a business process. However, the XPDL data exported from BPMN tool such as Bizagi Modeler still have much information unnecessary for CAM such as location, size of the graphical object, etc. We propose using CIM as an intermediate tool to convert XPDL data to standard data of CAM. Table 2 shows the contribution of A.White on mapping from BPMN to XPDL and our work is mapping XPDL to CAM standard. In this approach, we do not map all graphical object of BPMN, we focus only on popular elements of BPMN. With long BPMN process, we propose use Sub-process object to limit the number of objects on one business process. This limitation we will discuss on section 6.

TABLE 2. THE MAPPING FROM BPMN TO CAM STANDARD.

BPMN graphical object	XPDL mapping	CAM standard
BPMN process	<WorkflowProcess>	<Views xmlns= “”>
Start event	<Activity> <Route/> </Activity>	<Predicate " " > <ObsRulesSet> </ObsRulesSet> </Predicate>
Task	<Activity> <Implementation> </Implementation> </Activities>	<Predicate " " > <ObsRulesSet> </ObsRulesSet> </Predicate>
Decision	<Activity> <TransitionRestrictio> <Split type="XOR"/> < TransitionRestrictio> </Activities>	<Predicate " " > <ObsRulesSet> </ObsRulesSet> </Predicate>
Transition	<Transition/>	<State/> to <State/>
Sub-process	<Activity> <Implementation> <SubFlow/> </Implementation> </Activities>	<State> <Evaluation/> <AdaptationRuleSet/> <State/>
Stop event	<Activity> <Route/> </Activity>	<Predicate " " > <ObsRulesSet> </ObsRulesSet> </Predicate>

To make the relationships between BPMN - XPDL and CIM clearer, we can see in below example of a business process that modeled by BPMN tool will be analyzing and mapping to CAM standard.

Example 1:
The Process starts with an order being received (as shown in Figure 3). The order data is then sent through a “Check availability” Task. Order data is passed to the “Ship article” Task if it were available on the system. Then the order data continue sent through an application “Financial settlement”. After that, the process was finished in “Payment received”.

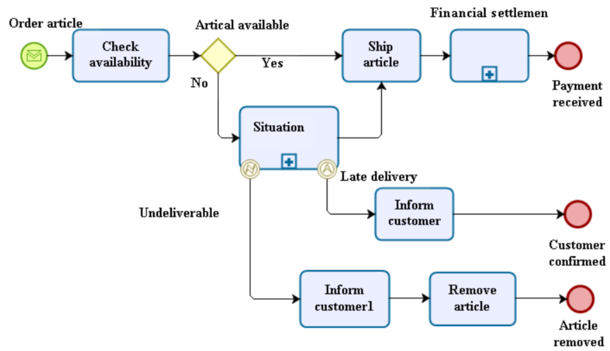

Fig. 3. E-Order process described by Bizagi Modeler.

If the Order not available, the order data sent through a “Situation” task. This task may generate a Process Error or Escalation, as shown by the Intermediate Events attached to the boundary of the task. A Task to Inform a customer follows the Process Error or Escalation Intermediate Event. If it has a problem with the process in this step, then a Task that uses an application to “Remove article” is performed. Order information is passed to action “Article removed” to complete the process. These details will be shown in the next section.

- E-Order process exported into XPDL file through Bizagi modeler tool:

```
<?xml version="1.0" encoding="utf-8"?>
 <WorkflowProcesses>
  <Activities>
    <Activity Id="8d711814" Name="Order article "/>
    <Activity Id="6b6a8fb3" Name="Check availability"/>
    …
  </Activities>
  <Transitions>
    <Transition Id="dcfc751e" From="8…4" To="6…3">
    <Transition Id="5…5" From="6…3" To="7…c">
    …
```

Translating to XML follow CAM's standard by CIM.

```
<?xml version="1.0" encoding="UTF-8"?>
<View xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchema="ViewSchema.xsd" >
<Description> Example view </Description>
<Predicate predicateId = "Order received "
EvaluationFrequency="0">
<ObservationRulesSet> </ObservationRulesSet>
</Predicate>
<Predicate predicateId = "Check availability"
EvaluationFrequency="0">
<ObservationRulesSet> </ObservationRulesSet>
</Predicate>
…
```

The result from CIM included Valid and Invalid State which described by a set of predicates. The detecting Invalid State is necessary to inform for an admin or user about the problem of the system. And maybe support a solution to solve the problem in some case or set an emergency situation.

Invalid and Valid state:

```
<State stateId="Invalid_2_Ma_00000000000010" >
<Evaluation>
<PredicateEvaluation predicateId = "Order received " value = "false"/>
<PredicateEvaluation predicateId = "Check availability" value = "false"/>
…
<PredicateEvaluation predicateId = "Err_Procureme" value = "true"/>
<PredicateEvaluation predicateId = "Financial settlement" value = "false"/>
</Evaluation>
…
<State stateId="Article removed" defaultState="true">
<Evaluation>
<PredicateEvaluation predicateId = "Order received " value = "true"/>
…
```

B. *Valid and Invalid states detecting.*

We use a special view with N predicate to analyze information form of each state at run-time. When the AAS system work in real life, the result of context's situation has not only Valid state but also Invalid state. With Valid state is the normal state which application should pass and complete one function or necessary step of the system operation processing. Invalid state is unnormal state by nature or non-conforming state in the natural order of execution. The Invalid state was created when the system has a problem with a sensor or error on collecting information from context during observation time. The CAM need manages both Valid and Invalid state to provide exactly adaptation rules for AAS. Especially, the situation of Invalid state can change or keep in many circles of AAS. We need to know exactly what happens with the system through the Invalid state situation to have suitable action. In this study, we propose a solution to detect and react with the Invalid state at run-time as shown in below example.

Example 2:
- View V: n predicate (nP) → 2^n States → S_i ($i = 1$ to 2^n) → R_i (R_i is set of adaptation rules in S_i), with "S_i may be Valid State or Invalid State". We suppose to use two counter C and Cd to detect the changing of Invalid State. The main goal of this detecting is keeping or changing the previous state to have suitable Adaptation rules as shown in Figure 4.

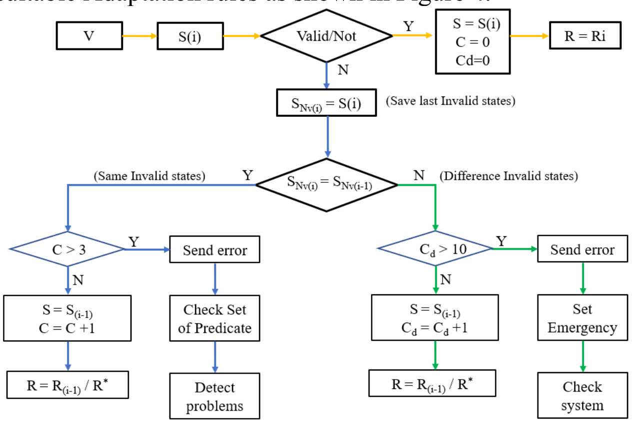

Fig. 4. Detection processing Invalid states.

- In case of Invalid state repeat many times:

+ If only one Invalid state appearance on many observation circles: Add a Counter (C) of each Invalid state, if situation of the state doesn't change in next circle of observation then $C = C + 1$; Check Counter: if $C > 3$ (propose)→ Send command Error to user → Check "Set of Predicate in this state" →Detects problem of system.

+ If many Invalid states appears random: Set a Counter of random Invalid state (C_d) and increase C_d after each Invalid state. If $C_d > 10$ (propose) → Send Error and set Emergency state: wait for checking system.

V. IMPLEMENTATION OF APPROACH

To implement our approach, we have some experiment with the WComp framework based on an auto-adapted system. It provides a prototyping and dynamic execution environment for Ambient Intelligence applications. The WComp supports the decision and action to the application in run-time depending on situations of devices [25]. We also have some experiments with another performance model [11] that allowed to evaluate the responsiveness of the adaptation based on the number of adaptation rules deployed by the CAM. With the current CAM, we take experiments to detect Valid and Invalid state. It is necessary improving the mechanism to get exactly AR in case of Invalid state. In case of conflicts between adaptation rules with each situation of context in one view, we have experiment [26] that deploys all the relevant Adaptation rules on the Auto adaptive system and manage conflicts that could occur in the real operation of the system.

We also have several experiments with the idea of contextual observation chains that introduced by Rey [27]. Each predicate of views is an application modeled by the way of an assembly of devices. Combine with our WComp framework to build applications that able adapt to variations of the devices.

VI. DISCUSS LIMITED OF APPROACH AND SOLUTION PROPOSE.

In present, the current approach strongly links the state of a view with the adaptation rules to be deployed by CIM. However, each special view has different structural and type of data. It is not simple to the designer can link all features of each view into the standard of CAM. Example with BPMN, we can see that if BPMN process stored many tasks or condition block (maybe more than 10), we must oppose with a big number of Invalid state because we have 2^N state for N predicate. That makes XML program becomes long and difficult to designer manage program on testing and solve the error of data.

To solve limitations above, we propose using the intentional approach in building a description of state and adaptation rules. With long BPMN process, we need to support a limited number of the predicate for each process. The designer can combine some small tasks into one task block and write adaptation rules for this predicate such as the situation of one Sub-process.

Another limitation of this approach is the ability of conflict between adaptation rules when we combine more than two independent views in one application. Sometimes we have used the same predicate for two different views, that mean adaption rules of this view can change the state of the predicate in other views. That is another type of conflicts between views in AAS. In this case, we need to continue to experiment with other views and evolve our Context intermediate model according to these experiments. It is necessary to improve the description and action of the state to detect conflicts of AR at run-time.

ACKNOWLEDGMENT

We would like to thank you to all members of the Continuum Project [28] which collaborated on the design and implementation of the first version of our Context-aware management. We also thank EDF R&D and Nice Sophia Antipolis University for supported us through the U-Insither Project.

REFERENCES

[1] M. Weisser, "The computer for the twenty-first century," Sci. Am, 1991, 94-104.

[2] A. K. Dey, "Understanding and using context," Personal and Ubiquitous Computing, 2001, vol. 5, pp. 4–7.

[3] M. Miraoui, C. Tadj, C. B. Amar. "Architectural Survey of Context-Aware Systems in Pervasive Computing Environment," Ubiquitous Computing and Communication Journal, 2008, pp 68-76.

[4] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Micklas, A. Ranganathan, D. Riboni, "A survey of context modelling and reasoning techniques," Jounal of Pervasive and Mobile Computing 6, 2010, pp 161_180.

[5] S. Nešković, R. Matić, "Context Modeling based on Feature Models Expressed as Views on Ontologies via Mappings," Computer Science and Information Systems, 2015, pp 961–977.

[6] T. Chaari, D. Ejigu, F. Laforest, M. Scuturici, "Modeling and Using Context in Adapting Applications to Pervasive Environments," In the Proceedings of the IEEE International Conference on Pervasive Services, 2006, pp 111-120.

[7] A. Achilleos, K. Yang, N. Georgalas, "Context modelling and a context-aware framework for pervasive service creation: A model-driven approach," Pervasive and Mobile computing, 2010, 281-296.

[8] V. G. Motti, J. Vanderdonckt, "A computational framework for Context-aware adaptation of user interfaces," Proceedings of the Seventh International conference on research Challengs in Information Science, France, 2013, pp1-12.

[9] M. Terdjimi, L. M´edini, M. Mrissa, "Towards a Meta-model for Context in the Web of Things," Karlsruhe Service Summit Workshop, Feb 2016, Karlsruhe, Germany.

[10] P. D. Costa, "Architectural Support for Context-Aware Applications-From Context Models to Service Platforms," CTIT Ph.D-Thesis, 021(TI/FRS/021) The Netherlands University, Enschede, 2007.

[11] S. Lavirotte, G. Rey, G. Rocher, J. Y. Tigli. "A Generic Service Oriented Software Platform to Design Ambient Intelligent Systems," UbiComp/ISWC, ACM, Osaka, Japan, 2015, pp 281-284.

[12] G. Rey, T. C. Do, J. Tigli, S. Lavirotte, N. Thanh. (2016) "Intermediate Common Model The Solution to Separate Concerns and Responsiveness in Dynamic Context-Aware System," Journal of Computer and Communications Vol.5 No.4. pp 44-59.

[13] A. K. Dey, G. D. Abowd and D. Salber, "A Conceptual Framework and Toolkit for Sup-porting the Rapid Prototyping of Context-Aware Applications," Human Computer Interaction, 2001, pp 97-166.

[14] R. Laddaga "Self-Adaptive Software Problems and Pro-jects," The 2nd International IEEE Workshop on Soft-ware Evolvability, IEEE Computer Society, Washington, 2006, pp. 3-10.

[15] D. Ejigu, M. Scuturici, L. Brunie, "An ontology-based approach to context modelling and reasoning in pervasive computing," Pervasive Computing and Communications Workshops, 2007. NY, USA.

[16] B. Chidlovskii, "Using Regular Tree Automata as XML schemas," IEEE Advances in Digital Libraries. Washington, D, C. 2000.

[17] I. Jaouadi, B. D. Raoudha, B. A. Hanene, "Approach to Model-Based Development of Context-Aware Application," Journal of Computer and Communications, Published Online 2015 in SciRes, pp 212-219.

[18] D. Schumm, D. Karastoyanova, F. Leymann, J. Nitzsche, "On Visualizing and Modelling BPEL with BPMN," Workshops at the Grid and Pervasive Computing Conference. Switzerland, 2009.

[19] X. Li, M. Eckert, J. Martinez and G. Rubio, "Context Aware Middleware Architectures: Survey and Challenges," Sensors, 2015, pp 20570-20607; doi:10.3390/s150820570.

[20] N. Souchon, Q. Limbourg, and J. Vanderdonckt, "Task Modelling in Multiple Contexts of Use," Interactive System, 9th International Workshop DSV-IS, Germany, 2002. pp 68-82.

[21] S. H. Siadat, M. Song, "Understanding Requirement Engineering for Context-Aware Service-Based Applications," Journal of Software Engineering and Applications, 2012, pp 536-544.

[22] Gu, T., Pung, H.K. and Zhang, D.Q. "A Service-Oriented Middleware for Building Context-Aware Services," Network and Computer Applications, 2005, pp 1-18.

[23] Business Process Modeling Notation (BPMN) (2004). BPMN.org.

[24] Stephen A. White, "XPDL and BPMN," Workflow Handbook, Future Strategies Inc, www.wfmc.org, 2003, pp221-238.

[25] J. Y. Tigli, S. Lavirotte, G. Rey, V. Hourdin, D. C. Wo, E. Callegari, M. Riveill, "WComp middleware for ubiquitous computing: Aspects and composite event-based Web services," Ann.Telecommun, 2009, pp 197-214.

[26] S. Fathallah, S. Lavirotte, J. Y. Tigli, G. Rey, M. Riveill "The Dynamic Composition of Independent Adaptations including Interferences Management," Proceedings of the Seventh International Conference on Software Engineering Advances (ICSEA), Lisbon, Portugal, 2012, pp 18-23.

[27] G. Rey, J. Coutaz, "The Contextor Infrastructure for Context-Aware Computing," In Workshop Component-oriented Approaches to Context-aware Computing held ECOOP'04, Oslo, 14 June 2004.

[28] ANR Continuum. Programme VERSO, Continuum ANR-08-VERS-005, 12-2008/09-2012.