

Extending the UML 2 Activity Diagram with Business Process Goals and Performance Measures and the Mapping to BPEL *

Birgit Korherr and Beate List

Women's Postgraduate College for Internet Technologies
Institute of Software Technology and Interactive Systems
Vienna University of Technology
{korherr, list}@wit.tuwien.ac.at
<http://wit.tuwien.ac.at>

Abstract. The UML 2 Activity Diagram is designed for modelling business processes, but does not yet include any concepts for modelling process goals and their measures. We extend the UML 2 Activity Diagram with process goals and performance measures to make them conceptually visible. Additionally, we provide a mapping to BPEL to make the measures available for execution and monitoring. This profile and its mapping are tested with an example business process.

1 Introduction

Although business process performance measurement is an important topic in research and industry [5], current conceptual Business Process Modelling Languages (BPMLs) do not mirror these requirements by providing explicit modelling means for process goals and their measures [14]. Furthermore, the measures need to be integrated into the process execution and require continuous monitoring. The goal of this paper is to address these limitations by

- extending UML 2 Activity Diagrams with business process goals and performance measures to make them conceptually visible, and by
- mapping the performance measures onto the Business Process Execution Language (BPEL) to make them available for execution and monitoring.

Activity Diagrams are a part of the behavioural models of UML 2 [20] and are used for modelling business processes as well as for describing control flows in software. Activity Diagrams neither have quality nor quantity based elements to measure the performance of a business process. For instance, the modeller of a process has no possibility to express the maximum time limit for processing of a specific action - the basic element of Activity Diagrams - or a group of actions.

UML profiles are an extension mechanism for building UML models for particular domains or purposes [20]. We utilise this well-defined way to extend the UML 2

* This research has been funded by the Austrian Federal Ministry for Education, Science, and Culture, and the European Social Fund (ESF) under grant 31.963/46-VII/9/2002.

Activity Diagram with business process goals and performance measures. In a further step we define its mapping onto BPEL, and thus, provide the following contributions:

- The modelling of goals is a critical step in the creation of useful process models, as they allow a) to structure the process design, b) to evaluate the process design, c) to better understand the broader implication of the process design and, d) to evaluate the operating process [13]. This is made explicitly visible by the UML 2 profile (cf. Section 3).
- The UML 2 profile and its mapping onto BPEL enable the transformation of the business processes models developed in a UML modelling tool into BPEL. Thus, the conceptually described performance measures can be directly transformed into the execution language and can be used to monitor the process instances continuously.
- The business process models as well as the extensions based on the UML profile can be easily created, presented and edited with existing UML modelling tools, as almost all newer UML tools support UML profiles.

In the remainder of the paper, the role of business process goals and performance measures is briefly discussed (Section 2). As a foundation for the UML 2 profile, we have extended the UML 2 metamodel for Activity Diagrams in Section 3. This lightweight extension mechanism provides the concepts to present the business process goals and performance measures. In Section 4, we describe a set of OCL constraints of the UML 2 profile to indicate restrictions that belong to the metamodel. The UML 2 profile as well as the mapping to BPEL is tested with an example business process in Section 5 and Section 6, respectively. We close with related work (Section 7), future work (Section 8), and the conclusion (Section 9).

2 The Role of Goals and Measures in the Business Process

With business process reengineering Davenport, Hammer, and Champy created a new discipline at the beginning of the 1990ies and provided the theoretical background for business process modelling. So far, in the business process modelling community attention has only been given to the modelling of certain aspects of processes (e.g. roles, activities, interactions). These theoretical aspects are mirrored in several BPMLs, for example, in the Business Process Modelling Notation [4], the Event-driven Process Chain [21], the UML 2 Activity Diagram [20], etc. Kueng and Kawalek argued already in 1997 that little attention is paid to the value of making goals explicit [13]. Today, there are quite a lot of conceptual BPMLs available, but they still do not provide modelling means for business process goals and performance measures [14].

A business process is defined as a “group of tasks that together create a result of value to a customer” [7]. Its purpose is to offer each customer the right product or service, i.e., the right deliverable, with a high degree of performance measured against cost, longevity, service and quality [10]. Although process goals and performance measures are available in process theory, they lack the visibility in conceptual BPMLs.

According to [13], the modelling of goals is a critical step in the creation of useful process models for the following reasons:

- We need to be able to state what we want to achieve so that we are then able to define the necessary activities which a business process should encompass (i.e., goals are used to *structure the design*).
- A clear understanding of goals is essential in the management of selecting the best design alternative (i.e., goals are used to *evaluate the design*).
- A clear understanding of goals is essential to evaluate the operating quality of a business process (i.e., goals are used to *evaluate the operating process*).
- A clear expression of goals makes it easier to comprehend the organisational changes that must accompany a business process redesign (i.e., goals help the modeller to better understand the broader implication of design, beyond those of the business process itself).

For all the reasons described above, we capture the business process goals and represent them graphically in a conceptual BPML, namely the UML 2 Activity Diagram. Furthermore, Kueng and Kawalek recommend in [13] to define to which extent the process goals are fulfilled, to measure the achievement of goals either by qualitative or quantitative measures, and to define a target value for each measure. Target values are also very important for Service Level Agreements (SLAs) as well as for business process improvement. Harrington stated “Measurements are the key. If you cannot measure it, you cannot control it. If you cannot control it, you cannot manage it. If you cannot manage it, you cannot improve it.” [8]. In order to support Kueng’s and Kawalek’s statement, and all stages of Harrington’s statement, we need to integrate performance measures into conceptual BPMLs.

3 The UML 2 Profile

In this section, we describe the extended metamodel for Activity Diagrams for the UML 2 profile with business process goals and performance measures. Activity Diagrams are a part of the behavioural set of UML 2 diagrams, and are used for modelling business processes as well as for describing control flows in software. A UML 2 Activity Diagram specifies the control and data flow between different tasks, called actions, which are essential for the realisation of an activity. The UML 2 Activity Diagram currently does not support the graphical representation of business process goals and performance measures. Thus, it is not possible to show, e.g., time restrictions of the business process, its cost or quality requirements. UML offers a possibility to extend and adapt its metamodel to a specific area of application through the creation of profiles. This mechanism is called a light-weight extension. UML profiles are UML packages of the stereotype «profile». A profile can extend a metamodel or another profile [20] while preserving the syntax and semantic of existing UML elements. It adds elements which extend existing metaclasses. UML profiles consist of *stereotypes*, *constraints* and *tagged values*.

A stereotype is a model element defined by its name and by the base class(es) to which it is assigned. Base classes are usually metaclasses from the UML metamodel,

for instance the metaclass «*Class*», but can also be stereotypes from another profile. A stereotype can have its own notation, e.g. a special icon. Constraints are applied to stereotypes in order to indicate restrictions. They specify pre- or post conditions, invariants, etc., and must comply with the restrictions of the base class [20]. Constraints can be expressed in any language, such as programming languages or natural language. We use the Object Constraint Language (OCL) [19] in our profile, as it is more precise than natural language or pseudocode, and widely used in UML profiles. Tagged values are additional metaattributes assigned to a stereotype, specified as name-value pairs. They have a name and a type and can be used to attach arbitrary information to model elements.

Figure 1 illustrates a section of the UML metamodel for Activity Diagrams and its extension with stereotypes for representing business process goals and their performance measures. The triangle at associations marks the direction of reading of a relationship between the metaclasses to support the clarity of the metamodel.

The UML profile consists of four different stereotypes, namely «*Process Goal*», «*Measure*», «*Alert*» and «*Organisational Structure*». The stereotype «*Process Goal*» describes the specific intension of a business process and is quantified by at least one «*Measure*». The «*Process Goal*» extends the metaclass *Activity*, meaning that a «*Process Goal*» is described at activity level.

The stereotype «*Measure*» can be classified and implemented as «*Quality*», «*Cost*» and «*Cycle Time*» and extends the metaclasses *Activity Partition*, *Structured Activity Node*, and *Control Flow*. This means that the stereotype «*Measure*» can be described in three different ways. It is the modeller's role to choose the most suitable way to best describe a measure for a certain purpose, a user or user group. Moreover, the stereotype «*Measure*» is responsible for the concrete quantification of different goals as well as for measuring the performance of a business process. If the process is not performed according to the «*Measure*», an «*Alert*» is triggered.

A structured activity node has the function to group elements of an activity, in order to structure the activity [20]. A measure located in a structured activity node quantifies the section of the process that is covered. For instance, a structured activity node that is extended with the stereotype «*Cycle Time*» has to finish the processing of its actions within a certain period of time.

A measure positioned in an activity partition quantifies the section of the process that is covered by the role. According to the OMG [20], an activity partition identifies actions that have some characteristics in common. For example, if activity nodes have to be performed within a specific period of time, then they are grouped within an activity partition labelled with the stereotype «*Cycle Time*». It is also possible to nest the stereotypes. A stereotyped structured activity node labelled with «*Working Time*» can be nested in an activity partition, e.g., extending «*Cycle Time*».

A measure based on the control flow quantifies the cycle time, cost or quality between two actions. The OMG [20] defines a control flow as an edge that starts an activity node after the previous one is finished. For example, a control flow that is extended with the stereotype «*Cycle Time*» and connects two activity nodes, means that the stereotype measures a period of time the token requires from the activity node at the beginning of the edge to the activity node at the end of the edge.

The stereotypes «*Quality*», «*Cost*» and «*Cycle Time*» add more detail to the stereotype «*Measure*» and classify it. The stereotype «*Quality*» has the aim to

measure the quality of a business process, which can be expressed e.g., by a low number of complaints or a high customer satisfaction.

The stereotype «*Cost*» represents the financial expenses a business process requires e.g., for its execution. Its tagged values and operations are necessary to compute e.g. average values like the total and monthly average cost of a certain process. The performance measures of «*Quality*» and «*Cost*» are in contrast to the measures of the «*Cycle Time*» often more focused on the type level of a process, as the required data is often not available on instance level.

The stereotype «*Cycle Time*» presents a time based measure and defines the duration a business process instance, or part of it requires from the beginning until the end. The stereotype «*Cycle Time*» can be specialised as «*Working Time*» or «*Waiting Time*». «*Working Time*» presents the actual time a business process instance is being executed by a role. «*Waiting Time*» shows the time limit the process instance is allowed to delay further processing. Moreover, «*Cycle Time*» has two tagged values, for representing the target value and the actual value of the process duration or a part of it which is computed by an operation of the stereotype.

The stereotype «*Organisational Structure*» describes the different roles within an Activity Diagram, namely the «*Organisational Unit*» and the «*Organisational Role*». Furthermore, an «*Organisational Unit*» has at least one «*Organisational Role*». The purpose of these stereotypes is besides showing the role that performs certain actions, to make the «*Organisational Structure*» visible that is triggered by the stereotype «*Alert*», if an action or a group of actions is not executed within its performance measures. The stereotype «*Alert*» has two metaclasses, from which it is derived, one for time based measures, namely *AcceptTimeEventAction*, and one for non-time based measures, namely *AcceptEventAction*. An «*Alert*» belongs to exactly one «*Measure*» as well as to one element of the «*Organisational Structure*», and has one tagged value to show on instance level if an alert is caused or not.

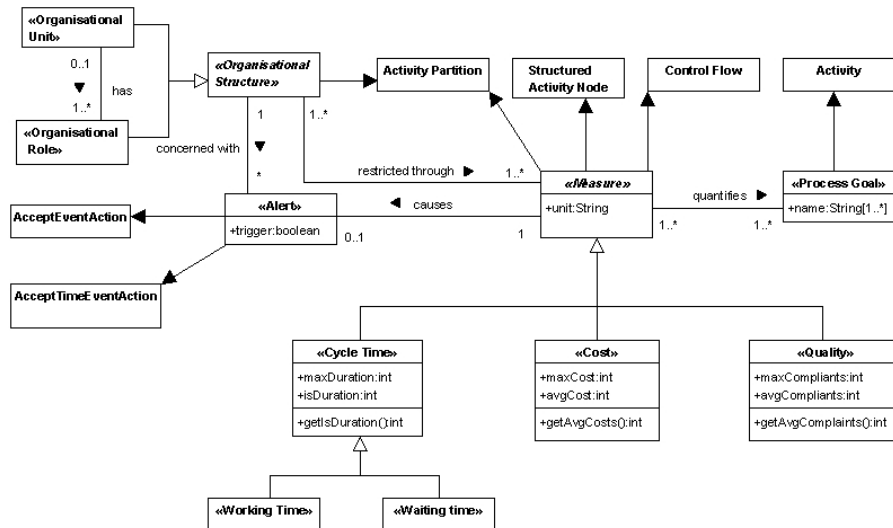


Fig. 1. Extended metamodel of the Activity Diagram for the UML 2 profile with business process goals and performance measures

4 Constraints

Constraints are applied to stereotypes in order to indicate restrictions. They specify pre- or post conditions, invariants, etc., and must comply with the restrictions of the base class [20]. Constraints can be expressed in any language, such as programming languages or natural language. We use the Object Constraint Language (OCL) [19] in our profile, as it is more precise than natural language or pseudocode, and widely used in UML profiles. Table 1 shows the OCL constraints with explanations in natural language for the stereotypes «Measure» and «Alert».

Table 1. OCL Constraints for the stereotypes «Measure» and «Alert»

Stereotype	Constraints
Measure	<p>If a measure is a cycle time based measure, then the occurring alert has the type of an AcceptTimeEventAction, otherwise the alert has the type of an AcceptEventAction.</p> <pre>context Measure inv: if CycleTime.ocIsKindOf(Measure) then Alert.ocIsKindOf(AcceptTimeEventAction) else Alert.ocIsKindOf(AcceptEventAction)</pre>
Alert	<p>If the actual value of the duration is higher then the maximum value of the duration, an alert will be generated.</p> <pre>context Alert inv: if cycleTime.isDuration > cycleTime.maxDuration then trigger = true else trigger = false endif</pre> <p>If the average cost is higher then the maximum cost, then an alert will be generated.</p> <pre>context Alert inv: if Cost.allInstances()->forAll(avgCost > maxCost) then Alert.trigger = true else Alert.trigger = false endif</pre> <p>If the average number of complaints is higher then the maximum number of compliants, an alert will be generated.</p> <pre>context Alert inv: if Quality.allInstances()->forAll(avgCompliants > maxCompliants) then Alert.trigger = true else Alert.trigger = false endif</pre>

5 Applying the UML 2 Profile to an Example Business Process

We demonstrate the practical applicability of the extension of the UML 2 Activity Diagram with business process goals and performance measures in Figure 1 with the example business process of an insurance company: the *Processing of Automobile Insurance Claims* business process (Fig. 2). We have refined the Activity Diagram by including a set of stereotypes, based on the various types of actions specified in the metamodels of actions in the UML superstructure in chapter 11 [20] and inspired by Bordbar et al. in [3].

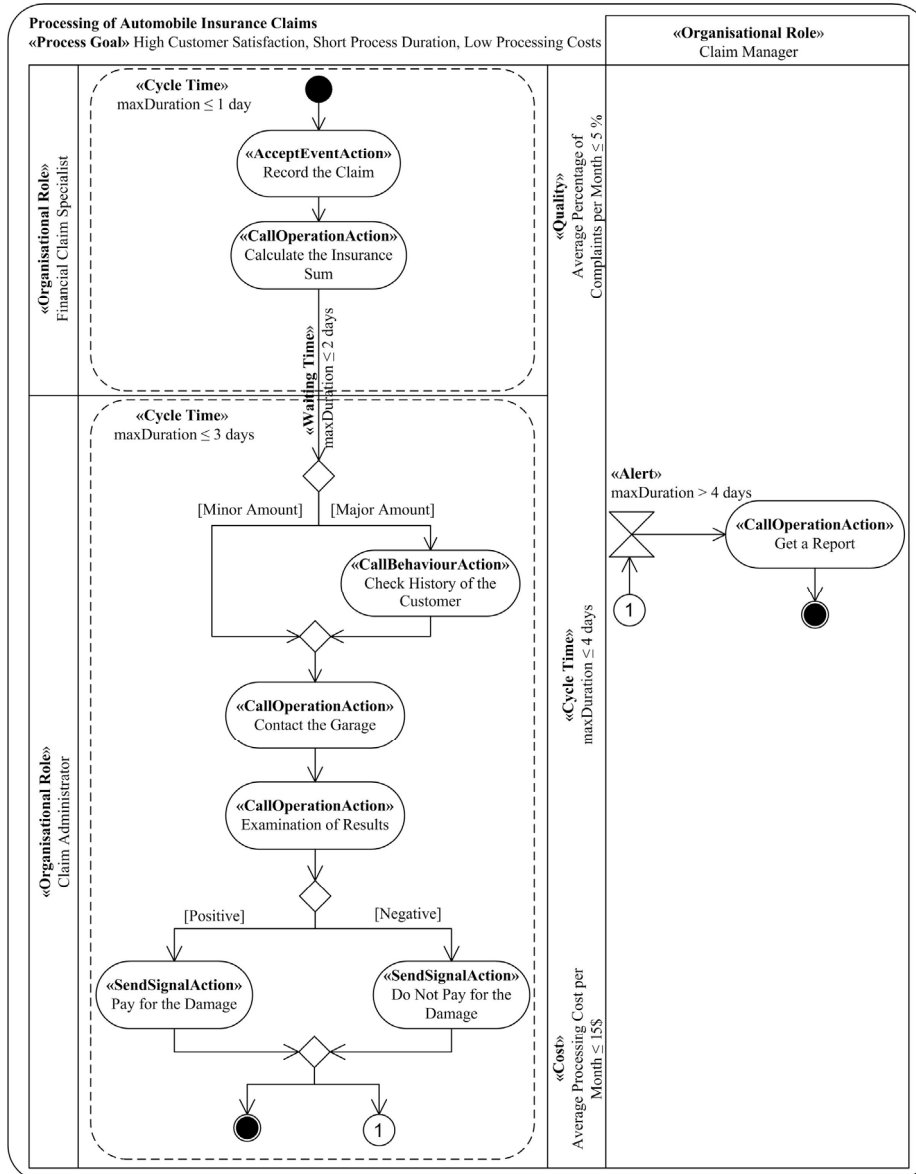


Fig. 2. Example business process based on the UML 2 profile for business process goals and performance measures

The overall goal of the processing of automobile insurance claims business process is to fulfil the «Process Goal» *High Customer Satisfaction, Short Process Duration and Low Processing Costs*. At the beginning of the process the «Organisation Role» *Financial Claim Specialist* is responsible for the actions *Record the Claim* and *Calculate the Insurance Sum*. The «Cycle Time» for these actions must not be more

than one day. This is shown in the structured activity node. After a «Waiting Time» of two days maximum that is illustrated on the control flow, the «Organisational Role» of the *Claim Administrator* has to follow up with the process. The claim administrator has a maximum «Cycle Time» of three days for processing its task. If the insurance sum is a major amount, then the claim administrator has to *Check History of the Customer* in the other case no action is required. After starting to *Contact the Garage* for the reparation, the *Examination of Results* has to begin. If the examination is positive, then the insurance has to *Pay for the Damage*, and the case is closed if the «Cycle Time» is not over for days.

The process has to meet the performance measures «Cost», «Cycle Time», and «Quality». The average processing cost per month has to be 15\$ maximum and the complaints should not exceed five percent. Only the «Cycle Time» is measured on process instance level. In the example, if the «Cycle Time» of the process is over four days, then the *Claim Manager* receives an alert and *Gets a Report* about that specific case and the business process terminates.

Figure 2 shows that a business process based on the UML 2 profile can be grasped at a glance. The extensions of the UML 2 Activity Diagram better illustrate the requirements of a certain business process and enhance the expressiveness of a model.

6 Mapping the UML Profile onto BPEL

The Business Process Execution Language (BPEL) is a language for specifying business process behaviour based on Web Services [9]. The UML profile will be mapped onto BPEL, in order to transform a specific business process modelling language and its conceptually described performance measures to an execution language as well as to make it possible to monitor the process instances continuously. Figure 3 shows the extended UML 2 Activity Diagram for BPEL based on [3]. Bordbar et al. present in [3] a transformation of the UML 2 Activity Diagram to BPEL to show the behavioural aspects of web services. We use this approach for mapping the different actions in Figure 2 to the BPEL tags in Figure 3. Furthermore, we are mapping «Cycle Time», «Waiting Time», «Working Time» by using the BPEL tag `<onAlarm>` as well as «Organisational Unit» and «Organisational Role» by using the `<partner link>` tag. We do not map the performance measures cost and quality, because we focus on the instance level of a business process and not on the type level. The web services a business process interacts with are modelled as `<partner links>` in BPEL, in the example business process this is the claim manager, the claim administrator as well as the financial claim specialist. Each partner link is characterised by a `partnerLinkType`, which we do not graphically show in the UML 2 Activity Diagram for BPEL. The tag `<onAlarm>` marks a timeout event, which is a part of the *Event Handler*. Both, the entire process and each scope can be linked with a set of event handlers. An alarm event goes off when the specified time or duration has been reached. The *for* attribute specifies the duration after which the event will be triggered. The alternative attribute *until* describes a specific point in time when the alarm will be fired. The clock for the duration starts at the point in time when the associated scope starts. In the example business process in Figure 3, we illustrate the

<onAlarm> tags. For the sake of simplicity we do not integrate the whole event handler into the diagram. Thus, we focus on one alarm event, executed by the role of the claim manager. If the overall process execution exceeds four days, then an alarm event will be created to inform the claim manager, and a report will be generated. Table 2 shows the mapping relations between the stereotypes of the UML 2 Profile and the BPEL tags, which are used in Figure 3.

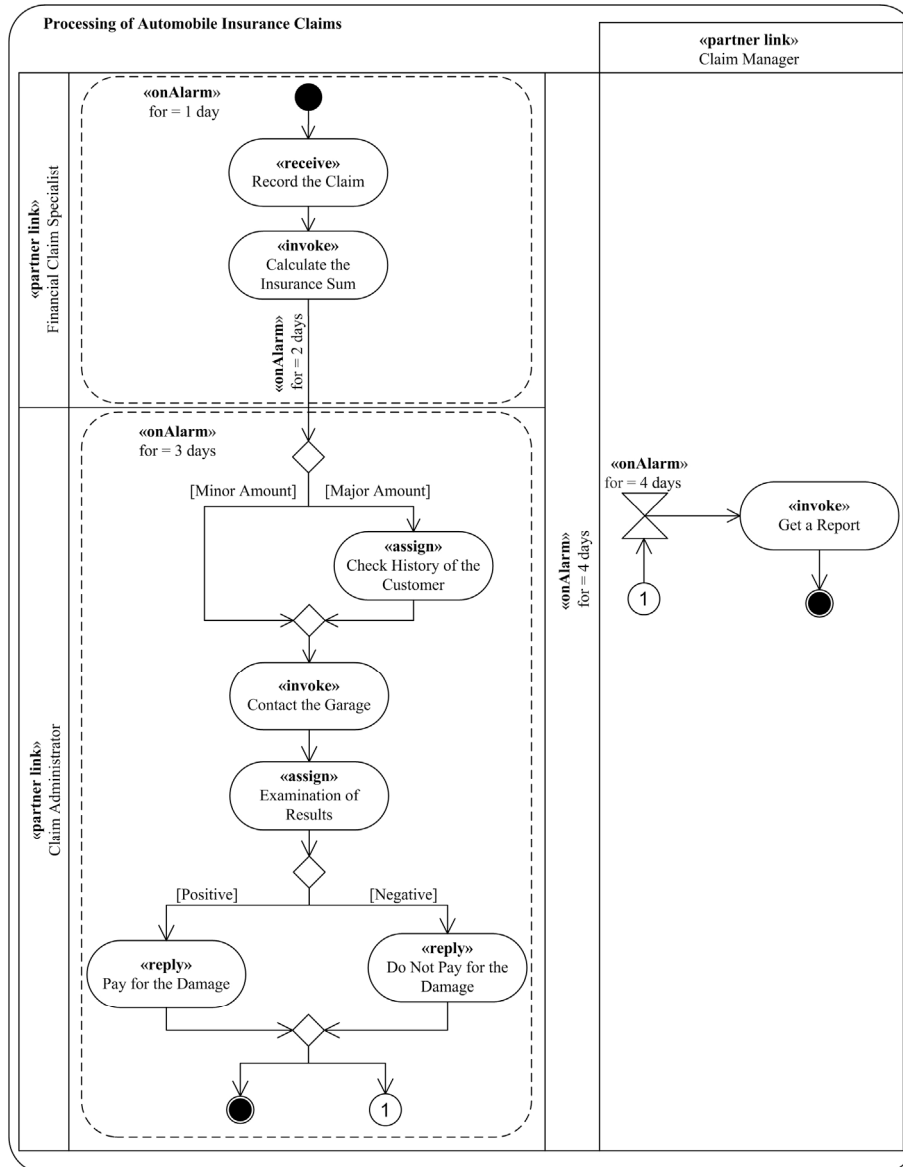


Fig. 3. Example business process based on the UML 2 profile for business process goals and performance measures and the mapping to BPEL

Table 2. Mapping relations between the UML 2 profile and BPEL

UML Base Class	UML Stereotype	BPEL Tag
Action	«AcceptEventAction»	«receive»
Action	«CallOperationAction»	«invoke»
Action	«CallBehaviourAction»	«assign»
Action	«SendSignalAction»	«reply»
Activity Partition / Structured Activity Node / Control Flow	«Cycle Time»	«onAlarm»
	«Waiting Time»	«onAlarm»
	«Working Time»	«onAlarm»
Activity Partition	«Organisational Unit»	«partner link»
Activity Partition	«Organisational Role»	«partner link»

7 Related Work

The related work consists of two parts. The first part is focused on different aspects of the quantification of performance measures, while the second part address to previous proposals for mapping a BPML to BPEL.

Aguilar et al. [1] developed a set of measures to evaluate the structural complexity of business process models on the conceptual level. The authors use the Business Process Modeling Notation (BPMN) [4] for their evaluation. The evaluation of performance measure like time or cost is not important for their work, the focus lies on measuring the core elements of BPMN.

The approach of Vitolins [22] is based on metamodelling according to the Meta Object Facility (MOF) [17] and aims to provide precise definitions of typical process measures for a UML 2 Activity Diagram-like notation. As a contrast to our work, the author annotates cost and time to each action separately as a note. There are no considerations to integrate the performance measures as graphical notation elements, although the approach lacks clarity and explicitness.

Nurcan et al. [15] adopted a goal-perspective, namely the map-driven process modelling approach to master the complexity of process modelling. The authors capture the strategic goals of the organisation as well as the tasks carried out by actors, to establish the importance of goals in process modelling.

There exist quite a lot of proposals for transforming / mapping UML 2 Activity Diagrams to BPEL, rather than for UML profiles to BPEL.

Bordbar et al. [3] present a transformation of the UML 2 Activity Diagram to BPEL to show the behavioural aspects of web services by using a metamodel, based on MOF [17] for BPEL. This work used OCL as a transformation language adapted from [12] at a time where no standard language for transformation definitions existed.

Gardner et al. [4] show a UML Profile for Automated Business Processes which enables BPEL processes to be modelled using an existing UML tool as well as a mapping to BPEL to automatically generate web service artefacts (BPEL, WSDL, XSD) from the UML profile. This work is rather out of date, because of the old UML version 1.4 and BPEL 1.0.

8 Future Work

In the sense of Model Driven Engineering (MDE) [16], the transformation of the Platform Independent Model (PIM), e.g. a UML profile to the Platform Specific Model (PSM), e.g. BPEL has to be the next step. Now the proper model transformation language has to be taken to find the way from the conceptual level to the implementation level. The most well-known transformation approaches [11] are the Query/Views/Transformation (QVT) approach [18], and the ATLAS Transformation Language [2]. The upcoming challenge is to create a MOF-compliant UML profile on metamodel level (M2), because both approaches, ATLAS and QVT operate in the M3 layered MOF-based metamodeling architecture [11], but the extension mechanism for creating profiles in UML is not a part of MOF.

9 Conclusion

In this paper, we have presented a UML 2 profile for integrating business process goals and performance measures into UML 2 Activity Diagrams. The profile provides an explicit illustration of the performance measures time, cost, and quality. Furthermore, it is possible to show the goals a business process must achieve, as well as the organisational structure that is concerned with alerts that belong to a measure. In order to capture these characteristics, we have extended the UML 2 metamodel for Activity Diagrams, and described them with stereotypes. Moreover, we have mapped the UML profile to BPEL, to transform a specific business process modelling language and its conceptually described performance measures into an execution language as well as to make it possible to monitor the process instances continuously. The UML profile and its mapping were tested with an example business process.

References

- [1] Aguilar, E. R., Ruiz, F., Garcia, F., Piattini M.: Evaluation Measures for Business Process Models, Proceedings of the 21st ACM Symposium on Applied Computing (SAC'06), April, Dijon, France, ACM Press, 2006.
- [2] Bezivin, J.: On the Unification Power of Models, Software and System Modeling Journal, Vol. 4 No. 2, pp. 171-188, 2005.
- [3] Bordbar B., Staikopoulos A.: On Behavioural Model Transformation in Web Services, Proceedings of the ER 2004 Workshops CoMoGIS, COMWIM, ECDM, CoMoA, DGOV, and ECOMO, Shanghai, China 2004, Springer Press, 2004.
- [4] BPMI: Business Process Modelling Notation, Business Process Management Initiative, Specification v.1.0, 2004. (06/05/31)
- [5] Casati F.: Industry Trends in Business Process Management – Getting Ready for Prime Time, Proceedings of the 16th International Workshop on Database and Expert Systems Applications (DEXA 2005), First International Workshop on

Business Process Monitoring & Performance Management (BPMPM 2005), August 2005, Copenhagen, Denmark, IEEE Press.

- [6] Gardner, T., Amsden J., Griffin C., Iyengar S.: Draft UML 1.4 Profile for Automated Business Processes with a mapping to the BPEL 1.0. IBM alphaWorks (2003), http://www-128.ibm.com/developerworks/rational/library/content/04April/3103/3103_UMLProfileForBusinessProcesses1.1.pdf, (06/05/31)
- [7] Hammer, M.: Beyond Reengineering – How the process-centered organization is changing our work and our lives. Harper Collins Publishers, 1996.
- [8] Harrington, J.H.: Business Process Improvement – The breakthrough strategy for total quality, productivity, and competitiveness. McGraw-Hill, 1991.
- [9] IBM: Business Process Execution Language for Web Services version 1.1, <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>, (06/05/31)
- [10] Jacobson, I., Ericson, M., Jacobson, A.: The Object Advantage – Business Process Reengineering with Object Technology. ACM Press, Addison-Wesley Publishing, 1995.
- [11] Jouault F., Kurtev, I.: On the Architectural Alignment of ATL and QVT, Proceedings of the 21st ACM Symposium on Applied Computing (SAC'06), April, Dijon, France, ACM Press, 2006.
- [12] Kleppe, A., Warmer, J., Bast, W.: MDA Explained. The Model Driven Architecture: Practice and Promise, Addison-Wesley, April 2003.
- [13] Kueng P., Kawalek P.: Goal-based business process models: creation and evaluation, Business Process Management Journal, Vol. 3 No.1., pp. 17-38, MCB Press, 1997.
- [14] List B., Korherr B.: An Evaluation of Conceptual Business Process Modelling Languages, Proceedings of the 21st ACM Symposium on Applied Computing (SAC'06), April, Dijon, France, ACM Press, 2006.
- [15] Nurcan, S., Etien, A., Kaabi, R., Zoukar, I., Rolland, C.: A Strategy Driven Business Process Modelling Approach, Special issue of the Business Process Management Journal on "Goal-oriented business process modeling", Vol. 11 No. 6, pp. 628-649, Emerald, 2005.
- [16] Object Management Group, Inc.: MDA Guide V.1.0.1, <http://www.omg.org/cgi-bin/apps/doc?omg/03-06-01.pdf> (06/05/31)
- [17] Object Management Group, Inc.: MOF 2.0 Specification, <http://www.omg.org/cgi-bin/apps/doc?formal/06-01-01.pdf> (06/05/31)
- [18] Object Management Group, Inc.: MOF QVT Final Adopted Version, <http://www.omg.org/cgi-bin/apps/doc?ptc/05-11-01.pdf> (06/05/31)
- [19] Object Management Group, Inc.: OCL 2.0 Specification, <http://www.omg.org/cgi-bin/apps/doc?ptc/05-06-06.pdf> (06/05/31)
- [20] Object Management Group, Inc.: UML 2.0 Superstructure, <http://www.omg.org/cgi-bin/apps/doc?formal/05-07-04.pdf> (06/05/31)
- [21] Scheer, A.-W.: ARIS – Business Process Modeling. Springer Verlag, 1999.
- [22] Vitolins, V.: Business Process Measures, Proceedings of International Conference Baltic DB&IS 2004, June 2004, Riga, Latvia, Scientific Papers University of Latvia Vol. 673, University of Latvia, 2004.