

Workflow Management Coalition



The Workflow Management Coalition Specification

Workflow Management Coalition Workflow Standard

Business Process Analytics Format (BPAF)

Document Number WFMC-TC-1015

Document Status – Working

Editor: Michael zur Muehlen (mzurmuehlen@stevens.edu)

February 20, 2008

Version 2.0

DRAFT VERSION – FOR PUBLIC REVIEW

Copyright © 2008 The Workflow Management Coalition

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the Workflow Management Coalition except that reproduction, storage or transmission without permission is permitted if all copies of the publication (or portions thereof) produced thereby contain a notice that the Workflow Management Coalition and its members are the owners of the copyright therein.

Workflow Management Coalition
99 Derby Street, Suite 200
Hingham, MA 02043 USA
+1-781-923-1411 (t)
+1-781-735-0491 (f)
wfmc@wfmc.org
<http://www.wfmc.org>

The “WfMC” logo and “Workflow Management Coalition” name are service marks of the Workflow Management Coalition. Neither the Workflow Management Coalition nor any of its members make any warranty of any kind whatsoever, express or implied, with respect to the Specification, including as to non-infringement, merchantability or fitness for a particular purpose. This Specification is provided “as is”.

Table of Content

1. CHANGE HISTORY	4
1.1. ACKNOWLEDGEMENTS	4
2. AUDIENCE	4
3. PURPOSE.....	4
4. INTRODUCTION	5
4.1. CONFORMANCE	5
4.2. CWAD 1.1 COMPATIBILITY	6
4.3. REFERENCES.....	6
5. THE BPAF STATE MODEL	6
6. THE BPAF EVENT FORMAT	8
7. XML SCHEMA FOR BPAF EVENTS.....	9
8. FIGURES AND TABLES	11
8.1. FIGURES	11

1. Change History

Version 2.0 Public Review, 20. February 2009

- Editorial Changes to Text
- Changes to XML Schema based on WfMC Meeting Discussion

Version 2.0 R2, 8. December 2008

- Added State Model Graphics and Legend
- Added Information on Event Data Format
- Added Information on BPAF State Model

Version 2.0 R1, 23. April 2008

- Revised states for Audit Data Events
- Alignment with Process State Model, and WS-HumanTask

Version 2.0 – Editor: Michael zur Muehlen (mzurmuehlen@stevens.edu)

- First draft of new document: XML Schema for Audit Data Events
- Changed Version, Date and Address Information

Version 1.1, 22 September 1998

- Final published version of CWAD (Common Workflow Audit Data) Specification, pre-XML

Version 1.0, 1 November 1996

- First published version of CWAD (Common Workflow Audit Data)

1.1. Acknowledgements

BPAF 2.0 was a collaborative effort with contributions from many individuals. Michael zur Muehlen edited the specification. Keith Swenson, Robert Shapiro, Shane Gabie and Nathaniel Palmer significantly contributed to the specification.

2. Audience

The intended audience for this document is primarily vendor organizations who seek to implement the Business Process Analytics Format (BPAF) of the Workflow Management Coalition (WfMC). It may also be of interest to those seeking to assess conformance claims made by vendors for their products. Comments should be addressed to the Workflow Management Coalition.

3. Purpose

The support of this specification in business process management systems allows for the recording and analysis of consistent audit data across heterogeneous business process management systems. During the initialization and execution of a process instance, multiple events occur which may be of interest to a business, including events that relate to the instantiation and completion of process activities, internal process engine operations and other system and application functions. Using BPAF-based information, a business can determine what has occurred in the business operations

managed by a business process management system. BPAF is designed as an implementation-independent data format that enables the aggregation and correlation of audit data across multiple platforms. While we anticipate that the major sources for BPAF data will be business process management systems, the use of the standard is not limited to these systems and other information systems may publish events following the BPAF data structure to allow for easier integration with other process-related audit data.

The collection of audit data will be under the control of an Audit Data Policy. The purpose of the audit data policy is to provide behavioral control over the audit mechanism. Through such a policy the amount, granularity and types of audit data collected can be controlled.

We expect the audit information to be utilized for both analysis (Process Intelligence and Analytics) and the derivation of status information (Business Activity Monitoring and Dashboards). In addition the audit data could be needed as proof of execution, e.g. to satisfy regulatory requirements. The analysis tools will want this information presented in a consistent format, representing all events that occurred within a given set of criteria, such as *“how long did process instance “x” take?”*, *“which activities have been performed within a given process instance?”* The BPAF specification provides a consistent format for the representation of individual audit events and their collection.

The set of runtime events covered in this specification is based on specific state machines for processes and activities. These state machines are aligned with specifications such as Wf-XML and BPEL4People/WS-HumanTask. Individual process management products may support additional states beyond those represented in this document, and thus may produce additional runtime events.

The definition of a query interface for runtime information is outside the scope of this document. When it is necessary to know the current status of a given process instance, available runtime events can be searched for the process instance ID of the process in question. The last recorded runtime event can be used to determine the current state of the process instance (e.g. open.running). To understand the overall state of the process, the runtime information measured against the process definition can provide an indication of the true state.

The WfMC has identified five functional interfaces to a process or workflow service as part of its standardization program. This specification forms part of the documentation relating to “Interface five” - supporting Management and Audit functions. This interface includes an XML schema for process audit events.

4. Introduction

All keywords and terms used within this specification are based upon the WfMC Glossary, or terminology used by BPMN.

For the purpose of this document, the terms process definition, business process model, and workflow model are all considered to represent the same concept, and therefore, they are used interchangeably.

4.1. Conformance

A vendor can not claim conformance to this or any other WfMC specification unless specifically authorised to make that claim by the WfMC. WfMC grants this permission only upon the verification of the particular vendor’s implementation of the published specification, according to applicable test procedures defined by WfMC.

4.2. CWAD 1.1 Compatibility

BPAF version 2.0 supersedes CWAD 1.1 (WfMC-TC-1015). CWAD was developed prior to the publication of XML and uses a proprietary data model. BPAF 2.0 is a new specification and does not provide backwards compatibility with CWAD 2.0 or 1.0.

4.3. References

The following documents are associated with this document and should be used as a reference.

General background information:

- WfMC, Terminology & Glossary (WfMC-TC-1011)
- WfMC, Reference Model (WfMC-TC-1003)

5. The BPAF State Model

The Business Process Analytics Format is designed as an XML schema for audit events. These audit events typically represent the nature, time and context of a particular state change in either a business process management system or one of the processes it manages. BPAF relies on a single state model that is based on both the Wf-XML state model and the BPEL4People/WS-HumanTask state model, pictured below. The green (sub-)states indicate the path through the state model that a successfully executed process or activity would take, whereas the blue (sub-)states represent unsuccessful exits from a running process.

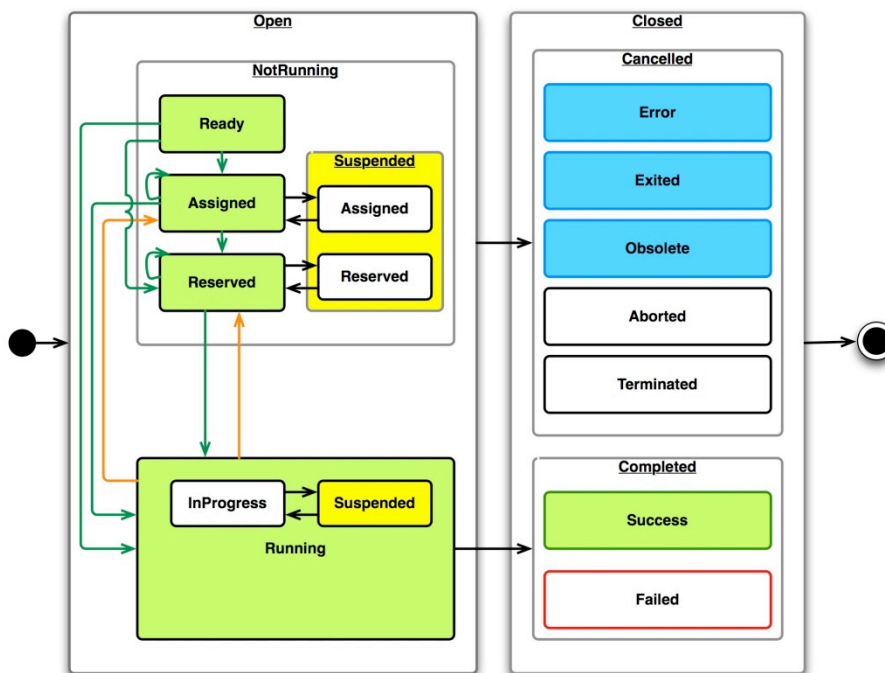


Figure 1: BPAF State Model

The BPAF state model supports multiple levels of nesting. A business process management system may support only the most basic states (OPEN and CLOSED), or it may refine these states into substates (such as OPEN.NOTRUNNING.SUSPENDED.ASSIGNED). This way a business process management system can extend the BPAF state model by introducing new substates to existing states. A system that

does not support these proprietary substates can reduce an even to the next closest superstate it understands (i.e. if an analytics tool does not understand `OPEN.NOTRUNNING.SUSPENDED.ASSIGNED` it may treat this event as belonging to the state `OPEN.NOTRUNNING.SUSPENDED`).

At the most basic level the BPAF state model has two states: `OPEN` and `CLOSED`.

- `OPEN`: An activity or process instance in the state `OPEN` can change state.
- `CLOSED`: An activity or process instance in the state `CLOSED` can no longer change state.

Metaphorically speaking, processes in the state `open` are alive, while processes in the state `closed` have terminated.

The states `OPEN` and `CLOSED` are divided into a number of substates:

- `OPEN.NOTRUNNING`: No active work is being performed on the process in this state. Time spent in this state would be recorded as idle time or wait time.
- `OPEN.RUNNING`: The process is actively being worked on. Time spent in this state would be recorded as processing time or work time.
- `CLOSED.CANCELLED`: The process has been forcefully terminated. Processes in this state have not achieved their objective.
- `CLOSED.COMPLETED`: The process has been fully executed. Processes in this state may or may not have achieved their objective.

The `OPEN.NOTRUNNING` state can be further divided to take into account human task management, in particular the allocation of work to particular performers and the deferred execution of tasks (e.g. when process participants can reserve work items on their personal worklist without actually starting the work).

- `OPEN.NOTRUNNING.READY`: The process/activity is ready to be processed but has not been assigned to a particular participant.
- `OPEN.NOTRUNNING.ASSIGNED`: The process/activity has been assigned to a role (potentially a group of participants), but work has not started yet.
- `OPEN.NOTRUNNING.RESERVED`: The process/activity has been assigned to a named user (a single participant), but work has not started yet.
- `OPEN.NOTRUNNING.SUSPENDED`: The process/activity has been moved into a hibernation state. Execution has not yet begun at this point, i.e. the process/activity cannot be executed from this state but has to be reactivated before execution can begin.
- `OPEN.RUNNING.SUSPENDED`: The process/activity has been moved into a hibernation state after its execution has begun.

The `OPEN.NOTRUNNING.SUSPENDED` state can be divided into further substates, depending on the state from which it was entered. If, for instance, an assigned task was suspended it may be recorded as being in the state `OPEN.NOTRUNNING.SUSPENDED.ASSIGNED`.

The states `CLOSED.CANCELLED` and `CLOSED.COMPLETED` can be further refined as well.

- `CLOSED.CANCELLED.ERROR`: The process/activity has ended due to an error in the execution.
- `CLOSED.CANCELLED.EXITED`: The process/activity has ended because it was manually exited prior to its completion
- `CLOSED.CANCELLED.OBSOLETE`: The process/activity has ended because it has been superseded

by a more recent version. This might occur if a process instance is migrated to a newer schema during the course of its execution and an activity that was part only of the old schema has become ready for execution but has not been executed yet.

- **CLOSED.CANCELLED.ABORTED:** The process/activity has been forcibly but gracefully ended, i.e. running activities and subprocesses were allowed to complete as scheduled before the end of the process instance.
- **CLOSED.CANCELLED.TERMINATED:** The process/activity has been forcibly ended, i.e. running activities and subprocesses were terminated before their scheduled completion.
- **CLOSED.COMPLETED.SUCCESS:** The process/activity has completed as planned and has achieved its objectives.
- **CLOSED.COMPLETED.FAILURE:** The process/activity has completed as planned but has not achieved its objectives.

6. The BPAF Event Format

Every state change in the BPAF state model can be represented as an XML message following the BPAF Event Format. This format has the following basic components:

General Information:

- **EventID:** A globally unique identifier for the individual event
- **Timestamp:** The time of the event occurrence
- **ServerID [optional]:** A globally unique identifier for the originating server of the event

Process Context:

- **ProcessDefinitionID:** The identifier of the process definition from which the current process instance has been derived.
- **ProcessInstanceID:** The identifier of the process instance that serves as the context of the event.
- **ProcessName [optional]:** The name of the process definition from which the current process instance has been derived.

Activity Context (only used if the event in question originated at the activity level):

- **ActivityDefinitionID [optional]:** The identifier of the activity definition from which the current activity instance has been derived.
- **ActivityInstanceID [optional]:** The identifier of the activity instance that serves as the context of the event.
- **ActivityName [optional]:** The name of the activity definition from which the current activity instance has been derived.

Event Detail Context:

- **CurrentState:** An identifier of the current state of the object that changed state, derived from the BPAF state model
- **PreviousState [optional]:** An identifier of the previous state of the object that changed state, derived from the BPAF state model
- **DataElement [multiple, optional]:** A name-value-pair that can be used to store additional process data that can later be used to correlate or aggregate events.

7. XML Schema for BPAF Events

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:bpaf="http://www.wfmc.org/2009/BPAF2.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wfmc.org/2009/BPAF2.0" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="Event">
    <xs:annotation>
      <xs:documentation>Business Process Analytics Format Event</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="EventDetails">
          <xs:complexType>
            <xs:attribute name="CurrentState" type="bpaf:State" use="required"/>
            <xs:attribute name="PreviousState" type="bpaf:State"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="DataElement" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="EventID" type="bpaf:ID" use="required"/>
      <xs:attribute name="ServerID" type="xs:NMTOKEN"/>
      <xs:attribute name="ProcessDefinitionID" type="xs:NMTOKEN" use="required"/>
      <xs:attribute name="ProcessInstanceID" type="xs:NMTOKEN" use="required"/>
      <xs:attribute name="ProcessName" type="xs:string"/>
      <xs:attribute name="ActivityDefinitionID" type="xs:NMTOKEN"/>
      <xs:attribute name="ActivityInstanceID" type="xs:NMTOKEN"/>
      <xs:attribute name="ActivityName" type="xs:string"/>
      <xs:attribute name="Timestamp" type="xs:dateTime" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="State">
    <xs:annotation>
      <xs:documentation>Enumeration of Possible Process and Activity States</xs:documentation>
    </xs:annotation>
    <xs:list>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="Open"/>
          <xs:enumeration value="Open.NotRunning"/>
          <xs:enumeration value="Open.NotRunning.Ready"/>
          <xs:enumeration value="Open.NotRunning.Assigned"/>
          <xs:enumeration value="Open.NotRunning.Reserved"/>
          <xs:enumeration value="Open.NotRunning.Suspended"/>
          <xs:enumeration value="Open.NotRunning.Suspended.Assigned"/>
          <xs:enumeration value="Open.NotRunning.Suspended.Reserved"/>
          <xs:enumeration value="Open.Running"/>
          <xs:enumeration value="Open.Running.InProgress"/>
          <xs:enumeration value="Open.Running.Suspended"/>
          <xs:enumeration value="Closed"/>
          <xs:enumeration value="Closed.Completed"/>
          <xs:enumeration value="Closed.Completed.Success"/>
          <xs:enumeration value="Closed.Completed.Failed"/>
          <xs:enumeration value="Closed.Cancelled"/>
          <xs:enumeration value="Closed.Cancelled.Exited"/>
          <xs:enumeration value="Closed.Cancelled.Error"/>
          <xs:enumeration value="Closed.Cancelled.Obsolete"/>
          <xs:enumeration value="Closed.Cancelled.Aborted"/>
          <xs:enumeration value="Closed.Cancelled.Terminated"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:list>
  </xs:simpleType>

```

```
<xs:simpleType name="ID">
  <xs:annotation>
    <xs:documentation>Namespace for Event Identifiers</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:NMTOKEN"/>
</xs:simpleType>
</xs:schema>
```

8. Figures and Tables

8.1. Figures

Figure 1: BPAF State Model 6