Towards a Semantic Framework for Business Activity Monitoring and Management

Claire Costello and Owen Molloy
Department of Information Technology
National University of Ireland, Galway
Ireland.

Abstract

Process performance management and continuous improvement initiatives can be significantly enhanced by real-time performance, quality and traceability information. For example, the focus of a Six Sigma quality programme is to reduce variability using statistical methods to highlight variance.

Current process modelling languages such as XPDL and BPEL provide little or no support for the inclusion of detailed process performance metrics. This paper describes a generic framework using event-based process modelling to support the definition and inclusion of performance metrics and targets within process models. The iWISE implementation of this framework is an XML and Web services-based infrastructure that uses this event-based model for integrating distributed processes and enhancing process visibility using real-time process metrics. Users can adjust alert thresholds on key process metrics in real-time. It uses an integrated rules engine, leveraging semantic technologies such as OWL and SWRL to write rules which are tested as process-related events occur in real-time.

Introduction and Motivations

Business Process Management (BPM) is the set of methods and tools required to manage the business processes making up an organization. A Six Sigma process improvement approach comprises five key phases: Define, Measure, Analyze, Improve and Control (DMAIC) (Adams, Gupta, & Wilson 2003). This life cycle comprises a set of phases which together form a closed loop of activities. The life cycle is depicted in fig. 1. The DMAIC approach can be summarized as follows:

Define Capture the process requirements in a definable and manageable format. Include in this definition key measurements required for analyzing process execution times or other related performance information.

Measure Continuously calculate key process metrics as processes are executing using event-based model defined in previous step. Metrics definitions are based on cycle times and Six Sigma calculations.

Analyze Analyze enterprise processes for critical changes based on acceptable limits for key parameters, for example, cycle time measurements exceeding a given target. Provide analysis using applicable tools or techniques such as correlation graphs, pareto charts and cause-and-effect (fishbone) diagrams. Web-based dashboard portal technologies are used as a central point for process monitoring and visualization.

Improve Use dashboards to identify bottlenecks and inefficiencies in the process and propose improvements. Simulate suggested process improvements to evaluate effect on process design and implement as approved.

Control Use control charts and other techniques to verify predictable process states.

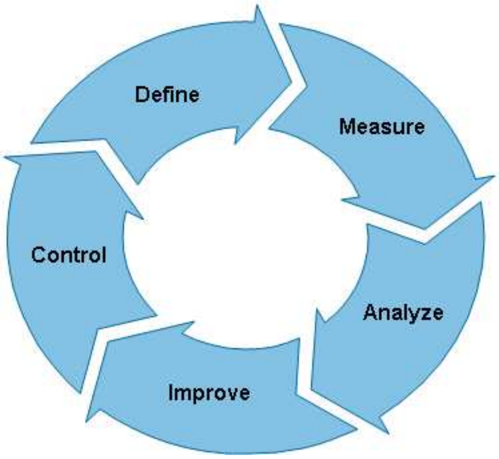

Figure 1: Phases of the DMAIC process improvement life cycle.

Each activity feeds into the next phase of the life cycle. It is difficult to maintain one model of a process where activities are managed by independent business functions. Lack of an integrated process view is one of the limitations of current Business Intelligence (BI) and reporting solutions (Morris, Vesset, & Fleming 2007). Processes, process metrics or Key Performance Indicators (KPIs), and even enterprise events are intrinsically related, but current practice is to handle them using separate software components. Such an approach leads to separation of definitions making it difficult to logically manage business events within their process context. Furthermore, the number of applications and databases used throughout an organization makes it difficult to create a single view of process performance, hence the need to create a common framework to describe process performance measures. (Sheina 2005) summarizes this by explaining that “highlighting the benefits of (near) real-time BI metrics and KPIs is one thing. But correlating them to process data is not a simple matter.” This paper presents a layered framework for process improvement that is based on a common underlying model.

The remainder of the paper is organized as follows. A brief overview of process performance management and various related technologies follows this introduction. Following this, an approach for integrating the definition of processes, events and metric thresholds for (near) real-time alerting using a rules-based approach is presented. Final sections cover the software designed to support this capability through leveraging a single model for process modelling, business event management and metric analysis and reporting.

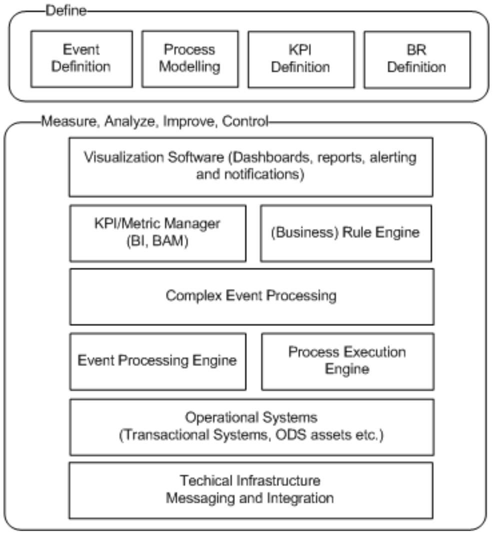
Define
Event Definition
Process Modelling
KPI Definition
BR Definition
Measure, Analyze, Improve, Control
Visualization Software (Dashboards, reports, alerting and notifications)
KPI/Metric Manager (BI, BAM)
(Business) Rule Engine
Complex Event Processing
Event Processing Engine
Process Execution Engine
Operational Systems (Transactional Systems, ODS assets etc.)
Techical Infrastructure Messaging and Integration

Figure 2: Methodologies and technologies ‘stack’ required for process, event and metric management.

Methodologies and Technologies

Methodologies and technologies currently used for creating the agile enterprise include BPM, Business Activity Monitoring (BAM), Event-driven Architecture (EDA), and a plethora of modelling and definition standards. In conjunction with these, Semantic technologies are a maturing research area which will also play an ever increasing role. The proposal is that a BAM framework for process improvement will leverage all of these technologies.

Figure 2 will serve as the basis for structuring a framework of contributing methodologies and technologies used to manage processes, events, and Key Performance Indicators (KPIs), or metrics, throughout this paper. It also highlights the fragmented nature of various methods or initiatives which could be consolidated. For example, metrics, whilst aligned to processes in management science literature, usually live independently from the process model as part of Data Warehousing (DW) projects. In addition, the framework is not exhaustive. Increasingly, Service Oriented Architectures (SOAs) are gaining prominence in practice as a stable approach to integration. The Operational Systems level represents the various systems supporting daily transactions and functions of the business. Examples of such systems are ERP, CRM and SC systems. An event-driven BI/BAM solution providing full process visibility is achieved using a full or partial implementation of these layers.

Process Performance Management

There are three levels of organization structure concerned with processes and metrics: strategic level, tactical level, and operational level. Strategic level goals are an aggregation of tactical level metrics. Tactical level metrics are an aggregation of operational level metrics (Junginger, Kuhn, & Bayer 2004). Key Performance Indicators can be classified into basic KPIs (for all levels) and aggregated KPIs (which are built on basic KPIs) (Junginger, Kuhn, & Bayer 2004). For example, at the strategic level, the business goal is *service leadership* in the market. Service leadership is in turn realized by many sub-goals, among them, *guaranteed service time*. A strategic KPI might be the ratio of the company’s cycle time to that of the market. At a tactical level, business processes are used to carry out business goals. Here, average cycle time KPIs for various steps in a complaints handling process will determine the performance of the company with respect to its goal to be a service leader. Operational systems used to manage the incident management process might include CRM and ERP software. KPIs at this level may be the number of incidents filed per day without errors. These KPIs are aggregated up from the operation layer, through the tactical layer to ultimately indicate if the overriding strategy goal is achieved. Companies can track their organizational effectiveness using this top-down approach to performance management (Melchert, Winter, & Klesse 2004). Process performance management is implemented using BI and BAM technologies.

Business Process Modelling

Central to each phase of the process life cycle in fig. 1 is a process model. Process modelling languages (PMLs) define notations for capturing business processes graphically. Business Process Modelling Notation (BPMN) and the Unified Modelling Language (UML) are examples of PMLs. Event-driven Process Chains (EPCs) model business functions, data and events. Events are created by processes or by external actors of the model. An EPC model does not have a machine processable representation. EPCs are used in the ARIS process platform. Integrated Definition (IDEF) is a set of modelling methods that can be used to capture business operations. Some PMLs have an underlying XML format, for example, BPMN. Process execution or definition languages (PELs/PDLs) are specifications understood by process execution engines. One such example is the Business Process Execution Language (BPEL).

(Mendling & Neumann 2005) provide a comparison of a set of process languages using 13 metamodel concepts. What emerges from this comparison is that support for defining or gathering metrics and data is not present. Without support for the measurement aspect of a process, monitoring processes remains an independent activity in the overall life cycle. A common model is necessary to support and link the various stages of a process life cycle (see fig. 1). In conclusion, current process modelling approaches lack a performance view required for activity monitoring and reporting.

EDA and CEP

Complex Event Processing and Event-driven Architectures, as the names suggest, are closely related. CEP cannot exist without an EDA infrastructure. However, an EDA infrastructure may exist where no CEP activities are performed on top of this. Event-driven architectures underpin BAM software. A complex event is a set of aggregated events. Event aggregation refers to the aggregation of sets or groups of lower-level events into a single higher level event that expresses the meaning of the lower-level events when taken together (Luckham 2002). A number of companies, for example, Syndera, are beginning to use such event processing capability as the integration layer in their frameworks. The RAPIDE event pattern language (Luckham 2002) is a declarative machine language that allows developers write event pattern rules. Any pattern matching software can use this language to support CEP applications.

As a new technology, EDA and CEP are lacking a formal definition process. In contrast, process modelling specifications have developed rapidly in recent years leading to overlapping efforts and the rise of de-facto standards as a result of backing from consortiums of large corporations. The OMG issued an RFI focussing on the standardization of Event definition and the relationship between EDA, CEP, BPM and SOA. Questions proposed include, amongst others:

- What is an event?
- What are the key/mandatory elements that describe an event?
- What are the major categories of events?
- Is there a standard Event ontology available/published?
- What enhancements could be made to existing modelling standards (ie. UML) to support modelling Event Driven Architectures?

Semantic Technologies

SWRL (Horrocks *et al.* 2004), a W3C member submission, is a Semantic Web Rule Language combining the Web Ontology Language (OWL) (W3C 2004) and RuleML (RuleML 2000). SWRL facilitates rule authoring using ontology concepts as part of rule predicates. In addition to OWL axioms, a knowledge base can now include rules written using a logical implication between an antecedent (body) and consequent (head). Since SWRL is based on the RuleML, it inherits the RuleML rule structure. With an appropriate event ontology defined, reasoning on events using the concepts of CEP and aggregated events is still an area open for research to determine business value.

Business Activity Monitoring

It is accepted that BI approaches provide an historical perspective on what has happened and do not provide insight into what is happening in an organization at any given moment (Nesamoney 2004) and (Golfarelli, Rizzi, & Cella 2004). BAM seeks to minimize decision latency by reducing the time between a business event occurring and a user making a decision about that event (Hackathorn 2002). BAM technology is an extension of application integration and messaging technologies that taps into transactions of IT systems (Knifsend & Debb 2005). To do so, BAM technology relies on a robust definition of both processes and events to deliver punctual information to its end users. A BAM system must be able to (Nesamoney 2004):

- detect events occurring in enterprise systems relevant to the activity being monitored;
- calculate information for temporal processing;
- integrate event and contextual business information on the fly delivering quality data;
- execute business rules to set thresholds according to key performance indicators and other business-specific triggers; and
- provide intuitive interfaces for presenting rules and metrics.

To this end, (DeFee 2004) describes four important modules required for a serious BAM solution: an event processing module, a process definition module, a monitoring module, and a visualization module. These modules are represented in fig. 2.

Related Research

(Thomas *et al.* 2005) describe a loosely-coupled architecture reliant upon a business process expressed in a process execution language such as BPEL. The architecture is agent-based and uses the Web Ontology Language (OWL) for describing performance criteria for business processes and individual activities. The illustrative example monitors the cycle time for workflow instances and details the metric for computing the cycle time for a given workflow instance. The paper is not clear how the agents know where to get information or map it to its semantic equivalent. It also does not show how other performance metrics may be supported. It explains the role of each agent type in monitoring a particular process for a given criteria, but does not show how to model the metric itself or how the parameters of a metric are mapped from operational business data.

Research at IBM has produced a technical framework that supports “sense and respond” business management and this work is detailed by (Kapoor *et al.* 2005). Their framework aims to provide business responsiveness or “the ability for businesses to quickly and effectively adapt to impending threats and opportunities." In its basic operation, the framework extracts and transforms data from supply chain applications, integrates the data to a data warehouse, manages the events firing from the data warehouse and produces timely information depending on the scenario of interest.

A Web service-based intelligent Decision Support System called the "Solution Manager Service" is described in (McGregor, Schiefer, & zur Muehlen 2006). The infrastructure supports a centralized repository for gathering run-time metrics for processes and analysing processes in near real-time. The Solution Manager Service consists of five main components all of which leverage a common data model for enterprise events. In particular, the Solution Builder component allows users to define metadata and business objectives for targeted business processes. The framework provides Web services to define other Web services from a performance measurement perspective and to log and analyse the enactment of Web services. The auditing mechanism is included within the BPEL process using new elements defined by BPEL's extensibility rules. The contribution here focuses on the ability to monitor Web service executions which is important given that many process modelling and definition languages are based on Web services.

(Haller & Oren 2006) describe an "intermediate ontology" that will act as a common mapping mechanism between the various internal and external process formats used by business systems. Their multi-metamodel process ontology (m3po) aims to unify the concepts of existing models and includes definitions for five key aspects of workflow modelling: functional and behavioural aspect, informational aspect, organizational aspect, operational aspect, and other orthogonal aspects (such as security or integrity constraints). However, the measurement aspect of the process is omitted. It would be important to describe quality-of-service (QoS) metrics or business level KPIs relating to measurement systems such as Six Sigma. However, such measurement systems or ability to define thresholds for KPIs are unsupported by this ontology and other workflow and process modelling and definition languages (Mendling & Neumann 2005), (Thomas *et al.* 2005).

RuleBAM uses business rules generated from high-level business policies in order to test conditions and determine appropriate actions. The framework consists of several tools and technologies for "real-time monitoring, exception handling and repair, alert and report infrastructure, process event infrastructure, monitor and configure agent deployment, [and] solution management decision support" (Jeng, Flaxer, & Kapoor 2004).

(McGregor 2002) suggests an amendment to the WfMC reference model that incorporates business performance monitoring information for use with the Balanced Scorecard (BSC). The general approach of this research is the inclusion of targets in a broader management sense for determining organizational performance, for example, "allow 24 hours for completion of this activity." Further research and development is necessary to supply information not only to process execution engines, but also to process monitoring engines.

Although the works detailed above outline various frameworks for process performance monitoring and management, they do not detail a process model that explicitly contains the elements for aiding process monitoring activities in (near) real-time. In addition, since a process model is captured at the define phase (see fig. 1), then a user should also be able to express important business parameter thresholds or control limits such as target cycle time or expected utilization level. The value of this research is that performance metrics and relevant metric thresholds should be incorporated into the process model.

An Integrated Model for Processes, Events and Metrics

The literature describes process modelling and process monitoring as separate activities. This research bridges these disciplines through definition of a process model supportive of performance measurements. Once defined, such a model can be deployed where rules may be invoked and alerts generated in near real-time when exceptional situations arise. The following sections describe the essential components of a model designed for usage in the framework in fig. 2.

Defining Models

The process model and related performance information uses XML as its canonical representation. Performance information captured will be addressed throughout this section. The model element is the root element of the process model and acts as a container for all other concepts. Figure 3 presents the XML element `Model`. Each model has a unique ID, `modelID`, for identification purposes. The `Name` and `Description` attributes contain general information about the model. A boolean attribute, `root`, indicates if this model is the root model of a given multi-level model. A multi-level model is illustrated in fig. 4. The hierarchy of models are connected through the definition of sub-models for a particular process. For example, process P_1 is defined to a greater level of detail in Model M_1. However, Model M_0 is not aware of Models M_1 or M_2. Models contain processes where a `Process` represents a business activity or step. The level of granularity is not an issue; a process could represent a collection of other processes or an atomic task. Processes are connected via `Transition` elements.

This research does not concern itself with controlling process executions or verifying the correctness of complex workflow structures, but instead aims to model and analyse as-is processes which may execute across many different systems. For this reason, complex control structures such as those described in (van der Aalst 2003) are not represented here.

Defining Processes

A process represents a business activity. The XML Schema for a process is given in fig. 5. The `ProcessID` attribute and the `Name` and `Description` elements are self-describing. Each process has a `ProcessOwner` which captures the business personnel responsible for the process definition. A process owner has typical attributes such as

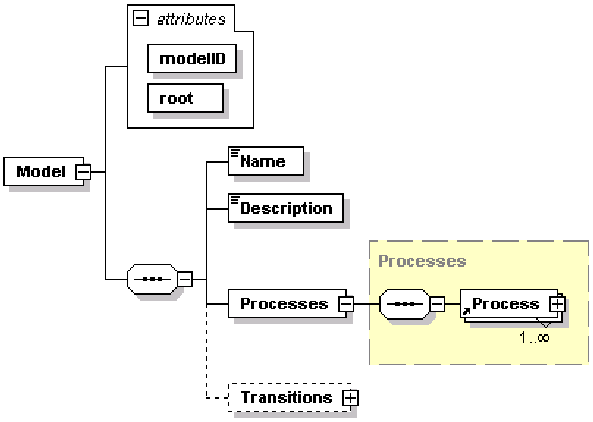

Figure 3: Model XML Schema.

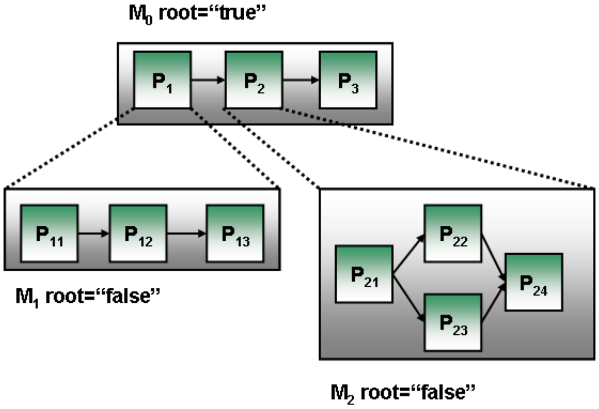

Figure 4: Relationship between models and processes in a multi-level model.

name, email address, and mobile number. Multiple events may be defined with a process and this taxonomy is described in the next section. A start and end event are mandatory event definitions for calculating total processing time for a process. As mentioned previously, multi-level modelling is supported. A process may have a sub-definition associated with it and this is recorded through a `SubModelID` element. There is a one-to-one relationship with a process and a sub-model. A model is not aware that it is a sub-model. The remaining attributes are related to the performance aspect of a business activity. Some elements are defined at process capture whilst others are updated as process statistics are calculated at runtime. The following set of elements are specified when the process is captured:

`CycleTimeUnits` Specifies the unit in which cycle time should be expressed, e.g., seconds, minutes, hours etc. Relevant in the context of `TargetCycleTime` for a process.

`TargetCycleTime` Specifies what the target cycle time (or total processing time) for a process is and is expressed in measurement units of `CycleTimeUnit`.

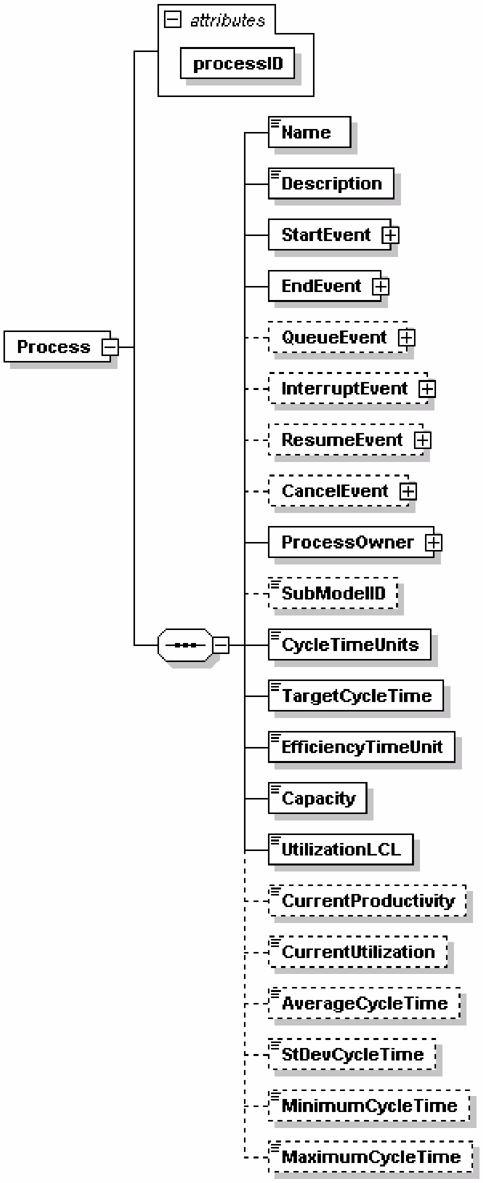

Figure 5: Process XML Schema.

`EfficiencyTimeUnit` Specifies the unit in which efficiency of a process is measured. This element is necessary when calculating process utilization and productivity statistics as a process is executing. Relevant to the `Capacity` and `UtilizationLCL` of a process.

`Capacity` Specifies the maximum throughput possible when a process is operating at its capacity. This element is necessary when calculating process utilization and productivity statistics as a process is executing.

`UtilizationLCL` Specifies the lower control limit (LCL) acceptable for process utilization statistics. This is expressed as a percentage.

The remaining set of elements are statistics gathered as a process is executing. These are calculated using the predefined set described above. The frequency of these calculations is an implementation issue and can be configured on the fly. In essence, these elements are calculated at runtime and represent an aggregate view of process execution for specified time intervals.

CurrentProductivity Records a current snapshot of how many units are being processed by a process for a given EfficiencyTimeUnit. This can also be expressed as the number of completed instances of a running process within a given time frame.

CurrentUtilization Records the current utilization level of a process as productivity expressed as a percentage of capacity for a pre-specified time unit. (See definition of EfficiencyTimeUnit, CurrentProductivity and Capacity.)

AverageCycleTime Records the average cycle time of a process for a given period of time.

StDevCycleTime Records the standard deviation of cycle time of a process for a given period of time.

MinimumCycleTime Records the least total processing time of a process for a given period of time.

MaximumCycleTime Records the highest total processing time of a process for a given period of time.

Defining Events

This modelling approach supports enterprise event modelling in conjunction with processes. General attributes of an event type include a unique ID, Name and Description. As part of this approach, the following classification of events has been identified:

- Queue Event
- Start Event
- Interrupt Event
- Resume Event
- Cancel Event
- End Event

This event classification scheme captures the various states of execution for a process. Each process must have an associated Queue, Start and End event definition. Each event definition follows the same structure and this is shown in fig. 6. Although each event type is defined using the same blueprint, its specialization is realized through its association with the correct process element of fig. 3. Therefore, an event type defined on its own is meaningless in its generalized form.

Business data manipulated in the context of a particular enterprise event occurring should be captured. For this reason, an XMLSchema is associated with an event definition. The XML Schema defines the format and structure of the business information linked to an event. For example, an Order Fulfilment Start Event will contain an XML Schema comprising Purchase Order information and other information relevant to the process.

Another important attribute of the event definition is XMLPathExpression and is required for event correlation purposes. For each process instance within an enterprise system, events are generated that contain information specific to that instance. In order to correctly assess a process, the correct events need to be grouped together. This event instance correlation mechanism is achieved using an element from the XML document to uniquely identify event instances. The same element must be present across all event definitions for each process definition within the same model. For example, for an Order Fulfilment process, the Order Number can be used to match start and end events. Since this information is included as part of the actual event business information, then it must be selected at event definition in order to correctly correlate the event instance data at runtime.

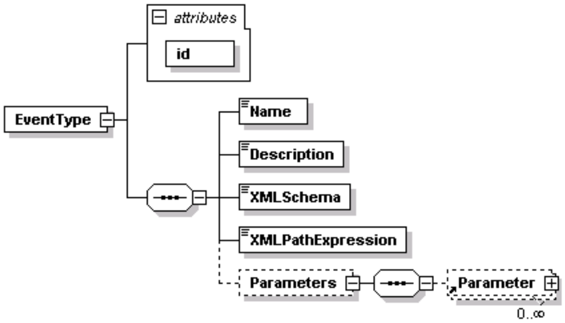

Figure 6: EventType XML Schema.

Defining Business Parameters

One element that is not expanded in fig. 6 is the Parameter element and its definition is given in fig. 8. Each event defined with a process can have multiple associated business parameters defined. This element is used to define business parameters not related to cycle time to allow calculation of Six Sigma type metrics on a constant basis. A parameter is a piece of business information selected from the business content packaged as part of an event at runtime. The business data is contained within the XMLPayload element in fig. 7. This payload is defined by the XMLSchema element in fig. 6. A user must define what the business data item of interest is at design time. In addition, a user must also specify an upper control limit (UCL) and/or a lower control limit (LCL). These pieces of data, defined as part of the event-based process model, can be used to drive the BAM solution.

The remaining optional elements are elements populated by a monitoring software component in real-time. Six Sigma ratios accounted for are Defects per Unit (DPU), Defects per Opportunity (DPO), Defects per Million Opportunities (DPMO), the quality level, and capability ratios.

SWRL Rules for Activity Monitoring

Using the event-based process model described in the previous section, an initial OWL ontology can be built (see fig. 9)

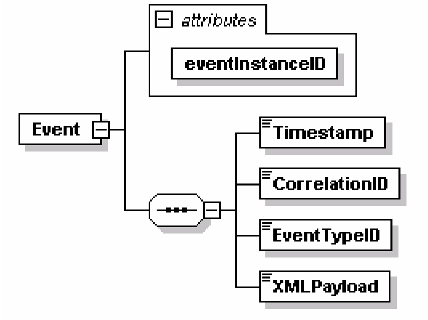

Figure 7: Runtime Event XML Schema.

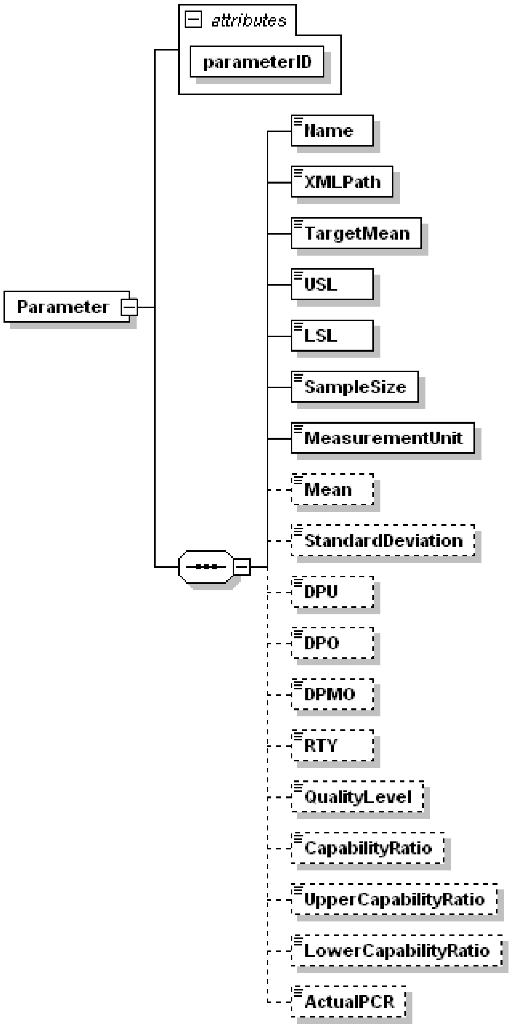

Figure 8: Parameter XML Schema.

which can serve as a foundation for specifying SWRL rules. In the context of BAM, SWRL rules for alerting have not yet been explored. As part of the framework in fig. 2, SWRL rules are defined around the process model during the *define* phase. After *measurement* has been performed by the KPI/Metric Manager, a Business Rule Engine may process the results to determine if user-defined thresholds have been exceeded and alerts can be generated to interested parties.

Table 1: OWL object properties.

Data Property	**Domain**	**Range**
hasBusinesObject	Process	BusinessObject
hasException	Process	Exception
hasPerformanceMetric	Process	Metric
hasStartEvent	Process	StartEvent
hasEndEvent	Process	EndEvent
hasQueueEvent	Process	QueueEvent

Table 2: OWL data properties.

Object Property	**Domain**	**Range**
hasTargetValue	Metric	Float
hasUpperControlLimit	Metric	Float
hasLowerControlLimit	Metric	Float
hasMetricValue	Metric	Float
hasBusinesObjectID	BusinessObject	String
hasTimestamp	Event	DateTime

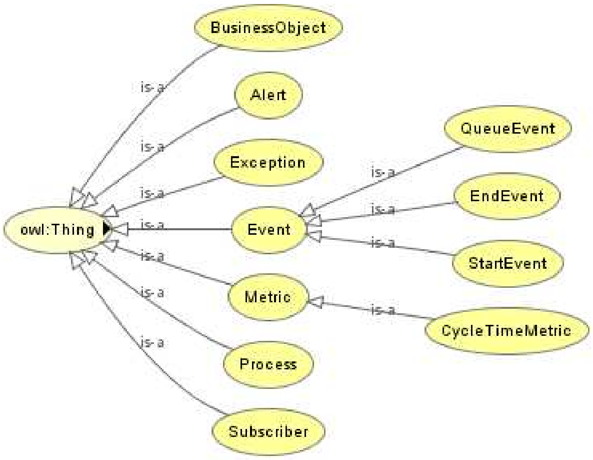

Figure 9: Defining concepts for a process metric ontology.

An experimental set of OWL object and data properties are defined and serve as the basis for defining the SWRL rules. A non-exhaustive list of object properties is given in table 1 to explore the concepts required for a BAM solution. In general, properties are based on the Process concept since process monitoring is the goal of the exercise. Table 2 lists some OWL data properties used when constructing SWRL rules. In the main, properties are related to either the Process or Metric OWL classes.

A sample SWRL rule for expressing when an exception has occurred with a given process is illustrated in table 3. This rule has been defined based on the ontology driven by the model definition. In this way, a user can define rules based on an event-based process model for the BAM soft-

Table 3: Abstract SWRL rule for monitoring values of business parameters.

Rule definition (abstract syntax)
Process(?x)ˆ hasBusinessObject(?x, ?b)ˆ hasPerformanceMetric(?x, ?y)ˆ hasMetricValue(?y, ?a)ˆ hasLowerControlLimit(?y, ?z)ˆ swrlb:lessThan(?a, ?z)ˆ ¬hasException(?x, ?b)

ware to evaluate at runtime. Once it is determined that a process exception has occurred, reporting software can package an alert for the appropriate users to view. This approach links the definition, measurement and monitoring phases of the process improvement life cycle together using a common model throughout.

iWISE Architecture for BAM

The iWISE software facilitates monitoring and improvement of business processes. The flow of activities in a typical iWISE system deploymentr the phases of the process improvement life cycle depicted in fig. 1 of the DMAIC Six Sigma methodology. Once a process is captured it is deployed to a process model management component. Once deployed, raw event streams are correlated with relevant processes to provide monitoring software with appropriate metrics. Given a (near) real-time process snapshot, personnel can change the process activity sequence to respond accordingly. iWISE is a common event infrastructure that uses the event-based process model described previously. The software components developed specifically for each of these capabilities and integrated by use of a common event and metric model are described in the following paragraphs.

The iWISE architecture is illustrated in fig. 10. Process models are captured using the iWISE Process Capture Tool (PCT) which is a Microsoft Visio-based standalone application. The PCT allows users to construct a process map and define all important enterprise events linked to a process. Users also specify metric thresholds such as target cycle time, capacity of a process, and efficiency time units for calculating utilization and productivity metrics. This flow is persisted using the event model described earlier.

The iWISE Legacy Listener components are configured to detect events in IT systems. Once detected, the events are constructed using the format in fig. 7 and sent to the iWISE Event Server where they are parsed and stored. Multiple listener components may be deployed within a business operating environment.

The iWISE Event Server is the central component responsible for managing models, event streams and metric calculations using the three software packages shown in fig. 10: the Model Manager, Event Manager and Metric Manager. The Model Manager receives process models from the iWISE PCT and compiles them for further processing. The Event Manager component receives enterprise events from Listeners defined in the format as specified during process capture. When raw events arrive, they are parsed and associated with the correct process. Enterprise events relating to particular process instances must also be correlated. Events are correlated using a unique identifier as determined at the process capture stage. The unique identifier can be a business object ID, for example, a purchase order number. As events are processed, the Metric Manager component generates metrics on-the-fly to provide an up-to-date process view for the iWISE Process Dashboard. It also performs metric calculations on-demand for the Process Dashboard. What results is the iWISE Operational Data Store built directly from a process map developed using the PCT.

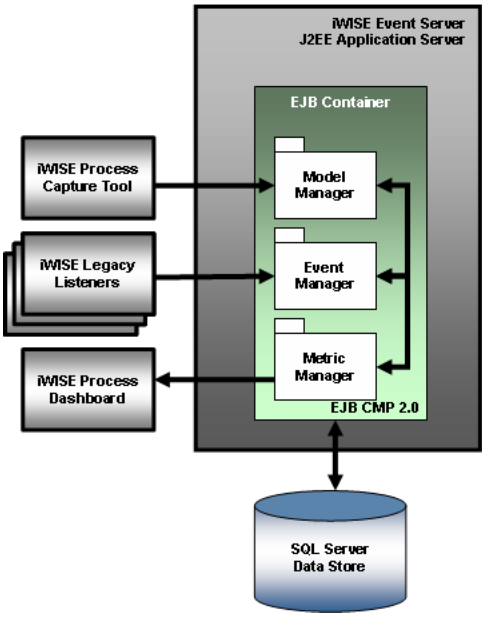

Figure 10: iWISE architecture containing the iWISE Event Server as the core.

The iWISE Process Dashboard is a Microsoft portal application that provides a timely snapshot of process performance. This component converses with the iWISE Event Server to generate interactive process maps with drill-down capabilities. Figure 11 shows the portal interface for a process model. Users can select process nodes and drill-down to the next level in the process hierarchy (if one exists) or they may choose to view process metrics and charts. Figure 12 depicts a full screen cycle time chart for a process. The target cycle time line is a value set at the process definition phase. The chart also shows the minimum and maximum values for the value added cycle time over the dates represented on the chart. The average value added cycle time is shown with a red point on the chart.

Generating Process Alerts

The iWISE Event Server Metric Manager component detects process execution anomalies in (near) real-time and

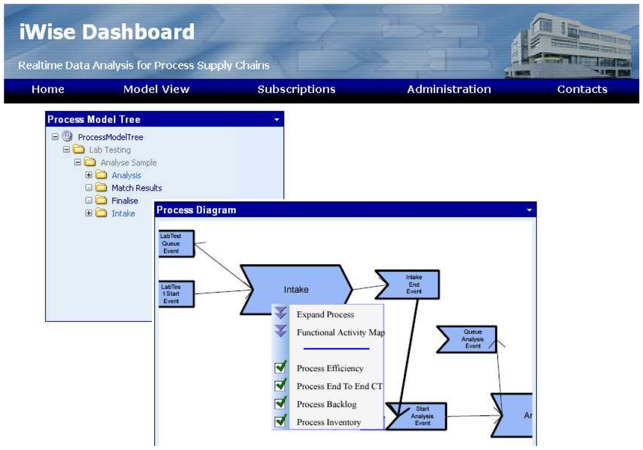

Figure 11: iWISE Dashboard Process View

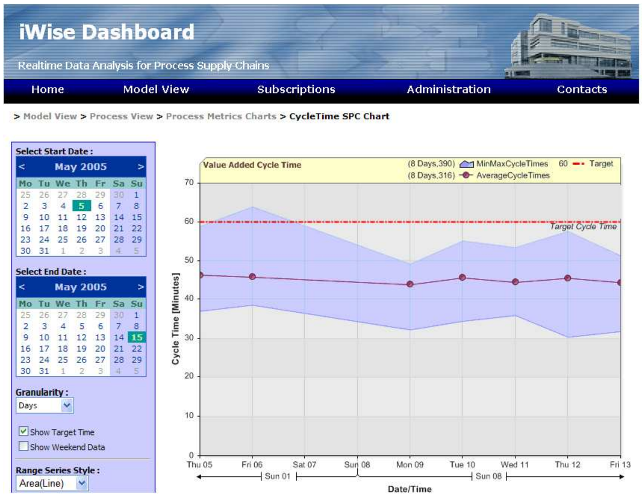

Figure 12: Sample iWISE process cycle time chart.

structures these exceptions for further processing by Microsoft Notification Services (MSNS).

Before alerts can be generated, rules must be defined based on the model described in this paper. The Protégé Ontology Editor is used to create both the ontology and SWRL rules to form the knowledge base required for analysing process runtime exceptions. Once specified, the rules are loaded into the Event Server for access at runtime (see fig. 13).

At runtime, a Java stateless session servlet (Process Monitoring Bean in fig. 13) is invoked at every time interval (set in configuration properties before application deployment) to calculate various process measurements. For example, using the rule definition given previously, the current process utilization level can be calculated and compared against a minimum acceptable value defined during the process design phase. The details of the process and metric are supplied to the Bossam reasoner (Jang & Sohn 2004) where the SWRL rules are used to reason if the metric supplied, and therefore the process, are outside normal limits of execution. In the case of process utilization, if the current level is below a minimum level, then that process is out of bounds. When a process metric is in such an exception state, the monitoring software will supply the exception information to an SQL Server database where MSNS will detect the information and generate a notification and alert if there are any subscribers defined for the process and metric in question. The Process Dashboard contains a process alerts subscription management page to allow users to subscribe to pre-defined alerts for processes deployed within the Event Server.

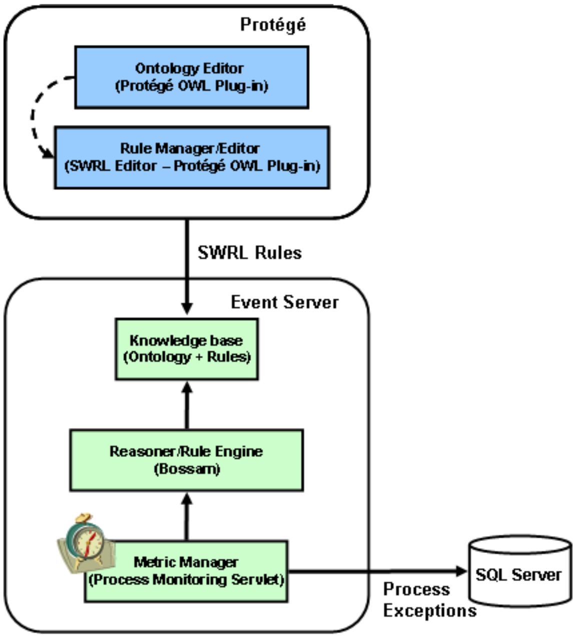

Figure 13: Detecting process exceptions.

At present, there are three rules defined which process facts generated at real-time to determine if a process is in an exception state. The first rule examines utilization of a process. During normal processing, the monitoring engine tells the Bossam engine facts relating to the current process utilization levels and proceeds to interrogate the engine to determine if the process has an exception. The rule for querying utilization levels is given in table 3. The two remaining rules relate to business parameter average values. These rules analyze target and average values for a process business parameter and, using the same processing sequence described, the monitoring engine can determine if the process has an exception. Any Six Sigma ratios or thresholds can similarly be monitored. This framework and solution does not attempt to define and implement an exhaustive set of SWRL rules for process monitoring, but instead takes the concept of measuring and defining thresholds to build a set of simple rules that may then demonstrate that semantic approaches can be utilized in this context. Interrogating a knowledge base of process and performance facts in real-time is possible given that rule sets can be authored and deployed with minimal operational disturbance.

Current Research Opportunities

The relationship between EDA and BPM is strengthening. In tandem with this, the combination of semantic technologies and BPM is the subject of increasing research. With respect to EDA, the development of event ontologies and standards to support event processing languages still remains an open area of research. By extension of this, the exploration of causality mining and enhanced event traceability will provide enhanced analysis techniques for process monitoring and management. The development of an event-driven architecture model leading to rapid deployments of custom solutions will reduce the costly software implementation turnaround time typically associated with such real-time monitoring projects. Potential also lies in the development of learning systems which can automatically detect causality in process performance data leading to early warning systems in, for example, the food and beverage supply chains.

Conclusions and Further Work

This paper described a model combining events, business processes, and metric information for use within a framework using EDA and Semantic approaches for Business Activity Monitoring. The advantage of this model lies in the fact that it brings together important business concepts necessary for a process improvement programme. Current work by the OMG around standardization of event understanding in relation to BPM and SOA technologies is on-going. This work makes some progress towards categorizing events and linking them to process definitions whilst also integrating a performance aspect to both process executions and the business data that is manipulated by events associated with those processes. Integrating BAM technologies such as EDA with Semantic approaches requires more research effort to fully understand the benefits to be gained.

The performance aspect of the process model described is by no means complete. The usage of OWL and SWRL can be further explored in terms of creation of a more extensive event and process performance ontology. The model defined here accounts for cycle time metrics and business data analysis with respect to numeric thresholds. Definition of these sets of metrics can be further enhanced and this research provides a basis for this work to continue to be explored. In any case, these three concepts are logically linked and therefore should be treated as an integrated model in a BAM framework.

The iWISE implementation of the framework discussed in this paper supports the full DMAIC cycle using a single event-based process view enhanced with metrics definitions and associated rules infrastructure. Multiple technologies were successfully integrated together to present a single view to the process designer and dashboard user.

References

Adams, C. W.; Gupta, P.; and Wilson, C. E. 2003. *Six Sigma Deployment*. Butterworth Heinmann of Elsevier Science.

DeFee, J. M. 2004. Business Activity Monitoring and Simulation. Electronically accessible: http://www.bptrends.com.

Golfarelli, M.; Rizzi, S.; and Cella, I. 2004. Beyond Data Warehousing: What's Next in Business Intelligence. In *7th ACM International Workshop on Data Warehousing and OLAP*, 1–6.

Hackathorn, R. 2002. Current Practices in Active Data Warehousing. Retrieved 14th September 2006 from http://www.bolder.com/pubs/NCR200211-ADW.pdf.

Haller, A., and Oren, E. 2006. A process ontology to represent semantics of different process and choreography metamodels. Electronically accessible: http://www.m3pe.org/deliverables/process-ontology.pdf.

Horrocks, I.; Patel-Schneider, P. F.; Boley, H.; Tabet, S.; Grosof, B.; and Dean, M. 2004. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Electronically accessible: http://www.w3.org/Submission/SWRL/.

Jang, M., and Sohn, J.-C. 2004. Bossam: An Extended Rule Engine for OWL Inferencing. In *Rules and Rule Markup Languages for the Semantic Web. Third International Workshop, RuleML 2004*, volume LNCS 3324/2004, 128–138. Hiroshima, Japan: Springer, Berlin.

Jeng, J.; Flaxer, D.; and Kapoor, S. 2004. RuleBAM: A Rule-Based Framework for Business Activity Management. In *SCC '04: Proceedings of the 2004 IEEE International Conference on Services Computing*, 262–270. Washington, DC, USA: IEEE Computer Society.

Junginger, S.; Kuhn, H.; and Bayer, F. 2004. Workflow-based Business Monitoring. *WfMC Workflow Handbook 2004*.

Kapoor, S.; Bhattacharya, K.; Buckley, S.; Chowdhary, P.; Ettl, M.; Katircioglu, K.; E.Mauch; and Phillips, L. 2005. A Technical Framework for Sense-and-Respond Business Management. *IBM Systems Journal* Vol. 44(No. 1):5–24.

Knifsend, F., and Debb, J. 2005. Using BAM to Empower the Organization: Unifying Events and Processes. *Business Integration Journal* 27–29.

Luckham, D. 2002. *The Power of Events*. Pearson Eduction Inc., Addison-Wesley. chapter Events, Timings and Causality, 87–112.

McGregor, C.; Schiefer, J.; and zur Muehlen, M. 2006. A Shareable Web Service Based Intelligent Descision Support System for On-Demand Business Process Management. *International Journal of Business Process Integration and Management* Vol. 1(Issue 3).

McGregor, C. 2002. *WfMC Workflow Handbook 2002*. Future Strategies Inc., Lighthouse Point, Florida. chapter The Impact of Business Performance Monitoring on WfMC Standards.

Melchert, F.; Winter, R.; and Klesse, M. 2004. Aligning Process Automation and Business Intelligence to Support Corporate Performance Management. In *The 10th Americas Conference on Information Systems*.

Mendling, J., and Neumann, G. 2005. *WfMC Workflow Handbook 2005*. Future Strategies Inc., Lighthouse Point, Florida. chapter A Comparison of XML Interchange Formats for Business Process Modelling, 185–198.

Morris, H.; Vesset, D.; and Fleming, M. 2007. Improving Business Performance Through Process Visibility. Whitepaper, IDC.

Nesamoney, D. 2004. BAM: Event-driven Business Intelligence for the Real-Time Enterprise. *DM Review* 14(3):38–40.

RuleML. 2000. The Rule Markup Initiative (RuleML). Online. Electronically accessible: `http://www.ruleml.org/`.

Sheina, M. 2005. BI and Business Process Convergence. *Computer Business Review Online*. Electronically accessible: `http://www.cbronline.com`.

Thomas, M.; Redmond, R.; Yoon, V.; and Singh, R. 2005. A semantic approach to monitor business process. *Communications of the ACM* 48(12):55–59.

van der Aalst, W. M. 2003. Workflow Patterns. *Distributed and Parallel Databases* 13(7):5–51.

W3C. 2004. Owl web ontology language overview. Online. Retrieved February 28 from `http://www.w3.org/TR/2004/REC-owl-features-20040210/`. Deborah L. McGuinness and Frank van Harmelen (eds).