iWISE: A FRAMEWORK FOR PROVIDING DISTRIBUTED PROCESS VISIBILITY USING AN EVENT-BASED PROCESS MODELLING APPROACH*

Claire Costello, Weston Fleming, Owen Molloy, Gerard Lyons and James Duggan
Department of Information Technology
National University of Ireland, Galway

Keywords: Business process performance management, Business Activity Monitoring (BAM), process modelling.

Abstract: Distributed business processes such as supply chain processes execute across heterogeneous systems and company boundaries. This research aims to provide an event-based process model to describe business processes spanning disparate enterprise systems. This model will be generic enough to support processes from multiple industry domains. In addition, this paper introduces the iWISE framework as a light-weight process diagnostic tool. The iWISE architecture uses the process model described to provide business process performance monitoring capabilities.

1 INTRODUCTION

Business processes set forth the steps required for achieving business goals. Advances in application integration technologies have resulted in inter and intra-business activities bridging many different systems. This factor has contributed to the menagerie of business processes executing at any given time within an organization. In addition, the fragmented nature of such business processes makes it difficult to monitor the health of business activities. Near real-time visibility will give business personnel the right information at the right time making businesses more responsive to changing business landscapes. Real-time visibility demands an architecture capable of continuous process monitoring. The iWISE project provides a common platform for integrating and monitoring extended business processes. It also provides process simulation and optimization functions.

The platform uses an event-based process model for describing business processes. Each event structure represents enterprise events and contains relevant business information. The integration of processes, events and business data provides a foundation for various process management activities such as monitoring, optimization and simulation.

The project's Industrial Advisory Panel includes leading Irish-based companies from two application domains. Advisors for supply chain integration include Pepsi Co. and Apple Computers. Advisors from the financial services domain include the Allied Irish Bank (AIB) Plc. and Voluntary Health Insurance (VHI). Partnership with these advisors ensures that project developments are applicable in real-world domains.

This paper describes related work and the iWISE process monitoring framework. Related research is discussed in Section 2. Section 3 provides a brief review of process modelling and monitoring. It also addresses the concept of Business Activity Management (BAM). Business process models from two different domains are presented in Section 4. These scenarios provide an understanding of the main modelling concepts required to model distributed processes. Section 5 describes the model used for defining and managing distributed business processes in this research and its contribution to current process modelling approaches. The iWISE framework for managing distributed process models and their runtime execution data is detailed in Section 6. Future work for iWISE is discussed in Section 7.

2 RELATED RESEARCH

Process management approaches and techniques are subject to continuous investigation by both academia and industry. Some of the trends in BPM include

*The support of the Informatics Research Unit of Enterprise Ireland is gratefully acknowledged.

Costello C., Fleming W., Molloy O., Lyons G. and Duggan J. (2006).
iWISE: A FRAMEWORK FOR PROVIDING DISTRIBUTED PROCESS VISIBILITY USING AN EVENT-BASED PROCESS MODELLING APPROACH.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - ISAS*, pages 224-233
DOI: 10.5220/0002497002240233
Copyright © SciTePress

Business Process Outsourcing, Business Process Intelligence and the use of Web service-based architectures to realize effective BPM solutions (Casati, 2005). The same author also describes a system currently being developed by Hewlett Packard (Palo Alto, CA.) which allows users model processes, services and metrics using a high level of business abstraction. This model then serves as a common model for using Extract, Transform and Load (ETL) components to observe events in the IT infrastructure in order calculate business-level metrics. The key element is the single source definition of processes, services and metrics to facilitate integration of business information to provide process visibility and impact analysis.

Research undertaken by the same institution (Casati and Discenza, 2000) developed a model for specifying interaction among workflow systems to achieve distributed business process interaction. The model extends traditional activity modelling approaches to include the definition of event nodes which represent points within the flow where events should be sent to or received from other processes. This approach focusses on the cooperation between distributed processes. By contrast, the aim of the research detailed by this paper is to provide an end-to-end view of processes using event-driven paradigms for process visibility.

Visualization of distributed processes, each with its own meta-model and semantics, is a complex task and the subject of inquiry in (Bobrik et al., 2005). The research does not aim to provide an execution framework for distributed business activities, but to provide a single view of a process spanning multiple heterogeneous systems to enable process monitoring. The challenge exists in translating multiple process meta-models into a common format for presenting the end-to-end process view to a user. The result of this work is a meta-model derived from the Workflow Management Coalition's XML Process Definition Language (XPDL).

Extending current process modelling approaches or specifications is not new. Data Warehouses (DWHs) provide a comprehensive resource of business data for supporting decision making processes within organizations. The extension of Event-driven Process Chains (EPCs) to include DWH information to provide a model that shows where and how business processes rely on DWH functionality is being explored (Stefanov and List, 2005).

Both Workflow Management Systems and Business Process Management Systems document the execution of process instances using workflow logs or audit trails. These resources can be quite hard to analyze directly. (Eder et al., 2002) details the development of a meta-model to describe the static and dynamic aspects as well as organizational aspects of a process. This meta-model serves as a basis for capturing processes in different representation techniques within a DWH environment. Since Data Warehouses are traditionally updated in batch-mode, real-time decision making is hindered by this gap in information availability. Current research examining the emergent synergies between process information and "Sense and Respond" architectures has resulted in the creation of near real-time process monitoring software. Work by (Kapoor et al., 2005) details such an approach to minimize decision making time for exceptions in processes.

Work carried out by (McGregor and Schiefer, 2004) aims to enable feedback on the organization's performance in a collaborative environment through analysis of Web service executions. The Solution Management Web Service (SMWS) framework provides Web services to define other Web services from a performance measurement perspective and to log and analyze the enactment of Web services.

Continuous process improvement relates to the ongoing analysis and redesign of processes where necessary to increase process quality and reliability. The Six Sigma approach to quality was developed originally by Motorola Inc. in the late 1980s to address the problem with variations in processes in order to achieve customer satisfaction. Quality may be defined as defect-free performance in all products and services (Smith, 1993). Despite being a rigorous methodology, Six Sigma is recognized as an effective quality programme that reduces process variation and costs (Caulcutt, 2001) and (Mitchell, 1992). The use of Six Sigma for process performance measurement is explored further in (Costello et al., 2005).

3 PROCESS MONITORING

Business Process Management (BPM) has evolved within the last decade to become a methodology for supporting a process-centred view of an organization. The process lifecycle comprises many phases: design, capture, simulate, execute, monitor, and optimize. Figure 1 illustrates the different layers of technologies and methodologies for managing the process lifecycle. At the top of the stack, process modelling and business content specifications aid the designer in capturing and specifying various parameters required for other stages of the process such as execution and monitoring. Business content specifications provide a common method for developing business information such as purchase orders and invoices that can be used for inter-company collaborations. The Universal Business Language (UBL) is an example of such a specification. The main output of these standards is a library of reusable components for building business

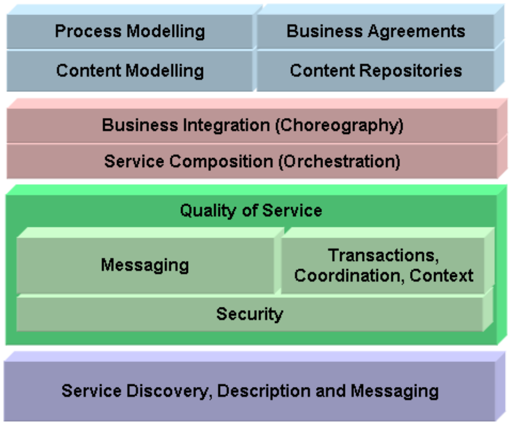

Figure 1: Modelling, deployment and integration technologies for process management.

documents.

Process modelling specifications define notations for capturing business processes. These standards typically provide templates and semantics for constructs such as processes (tasks or activities), process looping, sub-processes, normal flows, join and split flows, decision points, text annotations, pools and swimlanes. Examples of such standards include the Business Process Modelling Notation (BPMN) and the Unified Modelling Language (UML). Event-driven Process Chains (EPCs) model business functions, data and events. Events are created by processes or by external actors of the model. 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. Although created by the US Air Force for manufacturing environments, it has been adapted for other domains. Sixteen methods, from IDEF0 to IDEF14, are designed to capture various types of information through process modelling. IDEF0 to IDEF4 are the most commonly used methods. In particular, IDEF0 is used to model the functions of an enterprise by specifying the inputs, outputs, controls (constraints) and mechanisms (resources).

Business integration and service composition technologies detail the process from an execution point of view. These specifications are also known as process definition languages and such files are used as inputs to process execution engines. In the current technology landscape, this space is filled by Web services-based integration and composition languages. Examples of orchestration languages include the Business Process Execution Language (BPEL) and Business Process Modelling Language (BPML), whilst the Web Services Choreography Description Language (WS-CDL) is an example of a choreography language. The lower levels of Figure 1 specify the quality of service and service management technologies necessary to ensure successful implementation of business processes.

Business Activity Monitoring (BAM) is a process performance management technique that equips personnel at all echelons of an organization with timely information regarding process execution. This level of constant feedback helps make decisions more effective. BAM achieves this by capturing and processing events as they occur within the business IT environment. Indeed, BAM technology is an extension of application integration and messaging technologies that tap into transactions of IT systems (Knifsend and Debb, 2005). BAM technology relies on a robust definition of both processes and events to deliver such punctual information.

The iWISE Business Activity Monitoring methodology encompasses five steps: Define, Measure, Analyze, Improve and Control (DMAIC). This method follows from the Six Sigma methodology for process improvement (Adams et al., 2003).

Define Capture the process requirements in a definable and manageable format. Specify process name, owner, start and stop events. Identify processes that are major control points within the enterprise systems. Include in this definition key process or product parameters, acceptable parameter values, as well as measurements required for analyzing process execution times.

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 other 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. Use Web-based dashboard portal technologies as a central point of process monitoring.

Improve Use dashboard 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.

3.1 Modelling Distributed Business Processes

Business IT infrastructures typically include many enterprise systems. Each system is responsible for carrying out operations on behalf of its business users. It is difficult to model business processes that span multiple systems. What is required is an end-to-end definition of distributed business processes. This research combines the event-driven paradigms of messaging systems and process modelling techniques to provide a single view of such processes. Each process definition will be enriched with event and business object information which will, in turn, be usable in latter stages of the process lifecycle, such as measurement and analysis.

Business systems generate many events during daily processing. For example, an order fulfilment process is a sequence of activities required for managing orders received from customers. When a new order for a product is created, it is can be seen as an event occurring within the appropriate order system or database. The creation of the new order is a raw business event. The order fulfilment process describes the steps and control flow for managing orders, but these steps may involve the use of disparate systems. To create a single view of the process, it is necessary to capture appropriate raw business events in context of the process and create a model which defines these two entities cohesively. This unifies data-rich raw business events with process models describing the various business activities of an enterprise. The iWISE framework provides users with the appropriate software to describe the business activities and events. Before presenting a process model which can describe such distributed processes in terms of processes and events, two case studies from separate business domains are presented to illustrate the core modelling concepts identified for developing the process model.

4 ANALYSIS OF BUSINESS SCENARIOS

This section presents, diagrammatically, a comprehensive breakdown of two business scenarios into their core activities. These scenarios have been developed in collaboration with industry partners and represent fragmented business activities and the points of interaction with external parties. From the identified activities or tasks, a number of messages have been defined as important integration procedures in the supply chain context. The messages, their start and end points, and the potential business information have also been identified. The analysis of these two scenarios serves as a basis for the process model developed as part of this research.

The first scenario represents the interactions between a General Practitioner (GP) and a Hospital Laboratory (Lab) (see Figure 2). The second scenario is simplified view of a traditional supply chain scenario and depicts the activities for and interactions between the manufacturer and wholesaler (see Figure 3). Interactions between the wholesaler and the retailer have been omitted for clarity. In the case of both diagrams, each participants' set of activities are grouped using the swimlane notation of process modelling notations. The processes are modelled using traditional UML activity diagrams. The communication between each party is depicted using broken lines drawn between the appropriate process in each swimlane. This interaction also carries a message containing relevant business information. To fully understand the nature of these collaborative processes, a set of core modelling concepts have been determined: process, model, and event. These concepts are discussed in the next section.

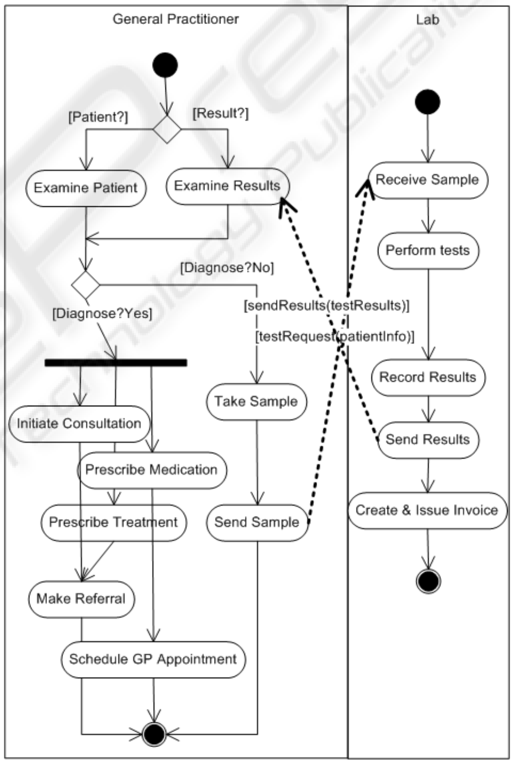

Figure 2: Collaborative process between GP and Lab.

5 EVENT-BASED PROCESS MODELLING

The process model developed as part of this research includes three important process artefacts: process, event and model. The process model also contains other artefacts for representing organizational structure charts such as organizational unit, role and partic-

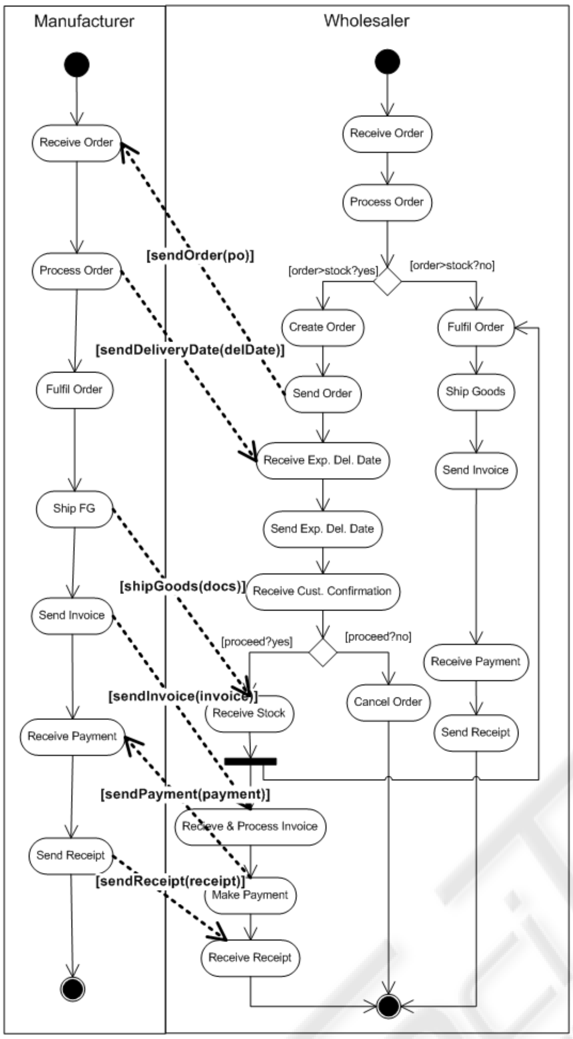

Figure 3: Collaborative process between Manufacturer and Wholesaler.

ipant. Each core artefact is explained in the following subsections. However, the model does not define control flow structures as used in existing modelling approaches. Distributed processes, when executed, may be controlled by local systems or local workflow systems. It is not the aim of this research to build a model that will control process executions (that is the role of a process execution engine), but to model and analyze as-is processes using raw system events. In addition, the model described here is XML-based. Each artefact has a corresponding XML Schema for construction and validation.

5.1 Processes

A process represents any step, task or unit of work within an organization. General attributes for a process includes a unique ID, process name and description. Organization employees may be responsible for a process and this information is also recorded with a process. A process can be associated with many events. Events are discussed in more detail in Section 5.3. A process may defined by another level of detail. This is captured using another model and is considered a level deeper in the model hierarchy. In Figure 4, process P1 has a sub-model M1. Process P2 has a sub-model M2. There is a one to one relationship between a process and a model to represent a sub-model of a process. Process P3 does not have a sub-model defined.

5.2 Models

At a diagram level, a model contains processes. Important attributes of a model include a unique ID, name, and description. An additional boolean attribute, called "root", denotes a model that is the root of a model hierarchy. Specifying models in this way is necessary for efficient retrieval and reconstruction of models containing processes. In Figure 4, model M0 is not aware of models M1 or M2. A model uses the concept of a "transition" to relate processes to each other.

5.3 Events

This modelling approach supports enterprise event modelling. An event represents something that happens in an enterprise where events occur as part of process execution. An event type definition is linked to a process definition (see Section 5.1). General attributes of an event type include a unique ID, name and description. There are six event types which can be defined and associated with processes.

- 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 event definition follows the same structure. In particular, an enterprise event will contain business information relating to the context in which the event occurred. For this reason, an XML Schema 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.

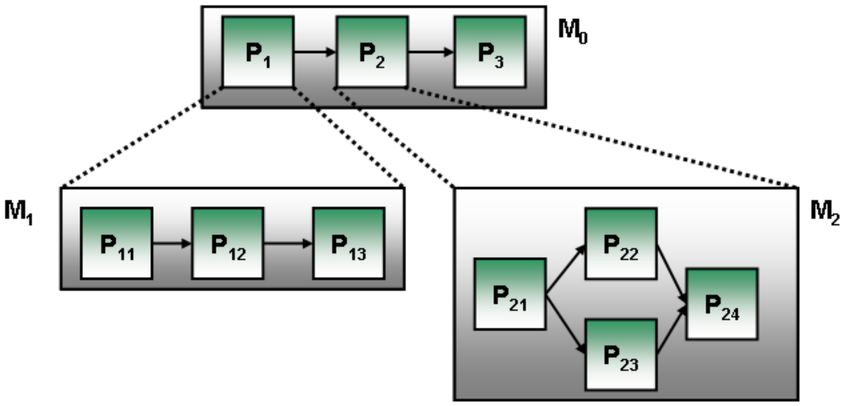

Figure 4: A model hierarchy.

Another important attribute of the event definition 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 matched together. This event instance correlation process 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. This event correlation attribute is called an "XMLPathExpression". 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.

Process models can be defined hierarchically. In this case, it is necessary to map events that occur at lower levels to their parent start and end events. In this case, the location of the parent event definition is specified. Finally, a "UnitsProduced" attribute indicates the output volume of the process each time it executes successfully. Each process definition must contain a Start Event and End Event definition. The remaining event types listed above are optional. This event classification scheme will be useful for calculating process cycle time measurements and analyzing process execution. Figure 5 illustrates the relationship between a process and events. Both "Order Entry" and "Order Fulfilment" have a start and end event. The "Order Entry" process has a Start Event (StartEvent1) and an End Event (EndEvent1). Each event carries XML data which encapsulates the business information associated with the event. This data is validated by the XML Schema specified by the event definition. The processes are connected via their events. The End Event (EndEvent1) of "Order Entry" is linked to the Start Event (StartEvent2) of "Order Fulfilment". A company boundary may be represented by the messages sent and received between its business partners. Any enterprise event may be modelled and associated with a process.

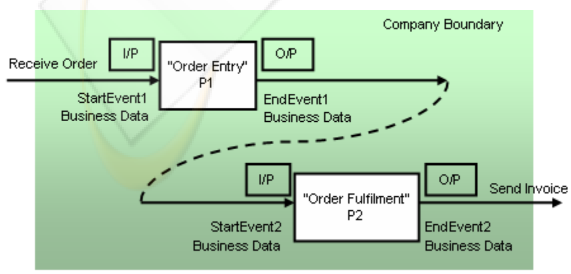

Figure 5: Enterprise process with associated start and end events.

This approach to modelling captures four essential views important in the context of a process model:

- The *data view*: captured using events which carry business data (XML data that conforms to an XML Schema). Using this approach, business data is linked to its creation and modification context within a process.
- The *functional view*: captured using the process box notation, which represents an activity in the organization at any level of abstraction.
- The *organizational view*: users can model the organizational structure and express relationships between units, roles and people. A process owner is an employee of the organization.
- The *control view*: processes are linked together via start and end events thereby preserving the flow between processes. These links are called transitions in the process model.

6 iWISE ARCHITECTURE FOR PROCESS VISIBILITY

The iWISE framework includes the following main components: the Process Capture Tool (PCT), Event Server, Process Dashboard Portal and Legacy Listeners (see Figure 6). The process capture software allows users to model distributed processes. Once captured, the process is deployed to the Event Server component. The Event Server is responsible for managing process model definitions and the business events and data generated by enterprise systems. Business process integration is achieved in two phases. The first phase, the process capture phase, requires users to capture the processes and events for a particular process model. The second phase, the process deployment phase, compiles the process model into objects required for event management. These steps fulfil the process design function of this framework.

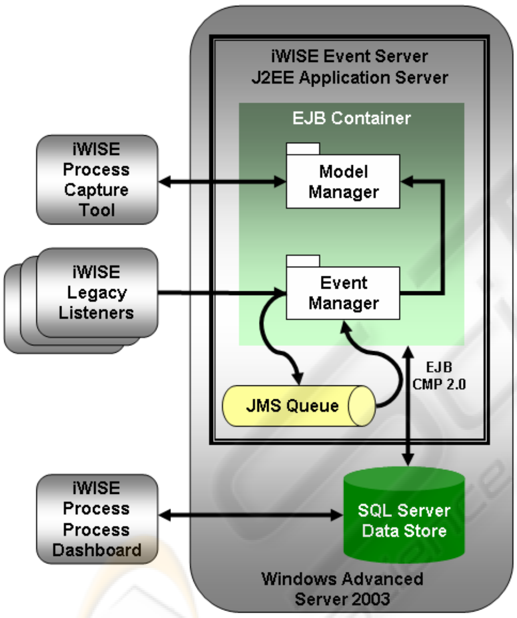

Figure 6: Event Server Components.

At runtime, the Event Server manages a continuous stream of raw enterprise events. The iWISE Process Dashboard has been developed using the .Net framework, and acts as the central portal and value chain dashboard. The process model created by the PCT is presented in the dashboard as a live process view with drill-down capabilities used to provide access to performance metrics generated by the Event Server. In addition, the Process Dashboard uses the event classification described previously to display current process execution statistics to the user, for example, number of processes started, cancelled, or completed.

6.1 Process Capture and Definition

In order to introduce a specific meta-model, which adequately defines an existing process structure to the system, one requires a software application to generate this model. For this purpose, the iWISE Process Capture Tool is being developed which is a GUI application that allows a user to visually construct and define their existing business process structure within a given supply chain. The PCT uses standard BPMN (White, 2004), a derivative of the UML based on (Eriksson and Penker, 2000) as visual contructs to define the overall process structure. The output of this activity is an XML-based meta-model which encapsulates and accurately represents this newly captured structure. This meta-model is validated against an abstract model schema and using Service Oriented Middleware is then persisted in a relational datastore. The tool allows for rapid definition of an existing process structure in a user friendly and intuitive environment, based on simple drag and drop actions of model components (see Figure 7).

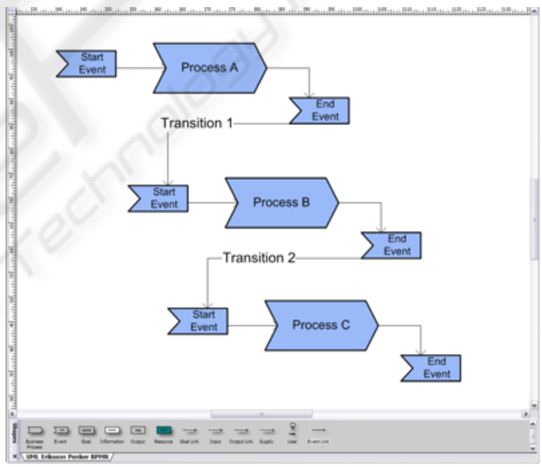

Figure 7: PCT interface for capturing an existing supply chain structure.

As the user constructs and modifies a process structure, the meta model will be generated/altered accordingly. During this phase the user will be prompted to define various business events within their organizational unit and attribute them to respective business processes. Furthermore, interesting parameters can be flagged from the business information these events are defined with for future monitoring. Also, transitions can be applied to business processes and their respective events which determine potential routes taken by products/payloads in a given supply chain. Along with the meta-model generated during the process capture phase, the PCT can produce graphical representations of the models in the Scalable Vector

Graphics (SVG) format and publish them to the web. This feature allows iWISE Process Dashboard components to retrieve these SVG models which serve as a base interface on which the dashboard can then extend in order to provide graphical-based supply chain monitoring capabilities. This feature enables the user to drill down through a process hierarchy and view the impact captured business events are having on the overall supply chain in real time.

The application was developed for the Microsoft .NET platform, and acts as a consumer participant in an extended Service Oriented Architecture (SOA). The PCT utilizes the flexibility of XML to manipulate and define business entities within a supply chain. Then using SOAP-RPC, the PCT is then able to invoke remote web services to handle entity persistence and pass native XML data structures as parameters (see Figure 8).

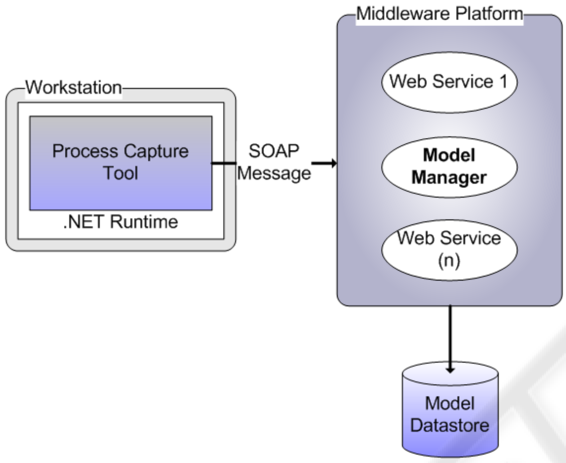

Figure 8: SOA context overview of the Process Capture Tool.

6.2 Process Deployment

The iWISE Event Server component is a Java-based enterprise application and performs two roles for process integration. Firstly, it manages process models captured using the PCT. Secondly, using the events defined as part of process models, it manages enterprise events received in near real-time. Each event has an event definition that is linked to a process definition (see Section 5.3). Figure 6 shows the Event Server architecture. Currently, the Event Server application comprises a Model Manager component and Event Manager component. The Model Manager receives models created using the PCT and persists the model definition using J2EE's Enterprise Java Beans (EJB) technology. Each model artefect described in Section 5 has its own corresponding entity bean structure.

The Event Server application is deployed within the J2EE-compliant JBoss Application Server. Interaction between the PCT and Model Manager software component is achieved using Web services technologies. The Web service methods receive models created by the PCT software in XML format. Data persistence for models and events is achieved using EJB Container Managed Persistence 2.0 with an underlying Microsoft SQL Server 2000 database (see Figure 6).

6.3 Event Management

The iWISE Event Server receives events from Legacy Listeners installed throughout the business environment. The listeners are responsible for responding to new system events, packaging them appropriately, and sending them to the Event Server for processing and storage. The Event Manager component manages a continuous stream of enterprise events using a combination of Web services and Java Message Queue (JMS) technologies. For each new event, the Event Manager looks up the event definition persisted by the Model Manager during model deployment. This allows the Event Manager to extract the correct data for event correlation purposes (see Section 5.3).

The principles of Event Driven Architectures (EDAs) are present in the design of the Event Server and Legacy Listener software. EDAs use events to communicate between loosely coupled software components. The Legacy Listener is configured to detect specific events occurring within the enterprise system and send them, in a format usable by the Event Manager software, to the Event Server. In this way, the Legacy Listener produces events and the Event Manager consumes them. EDA architectures facilitate agility since event-driven systems are designed for unpredictable and asynchronous business environments. The use of Web services as the transfer mechanism allows the addition of newly configured listeners for distributed event management. Using this approach, continuous data integration is achieved by managing the events received from Legacy Listeners located throughout the business context.

7 FUTURE WORK

The event-based process model introduced previously is a valuable source of information for Business Activity Monitoring software and other value-added services. A central operational datastore for process execution data and business information logged using the prescribed event format acts as a repository for process performance measurement and analysis activities. This wealth of information can be used to provide process visibility in near real-time. The Event Server consumes events as they are produced by the business systems. These events can be processed and further analyzed to derive events used to drive alerting and notification software. Processing of raw business events is known as Complex Event Processing (CEP). Such event-driven processing systems provide a flexible infrastructure for building real-time BAM software (Knifsend and Debb, 2005). Process Mining, (Agrawal et al., 1998) and (van der Aalst and Weijters, 2004), and statistical control techniques may also be applied to the iWISE Event Server repository to yield root cause analysis for out of control processes. In addition, processes can be simulated using iWISE simulation software which receives as input the event-based process model captured using the iWISE PCT. Using process analysis and simulation data processes can be optimized with minimal latency in a constantly changing business environment.

In recent years, consortia of businesses and research institutions have developed process modelling and definition languages to support interchangeable business processes. (Mendling et al., 2004) analyze fifteen of the currently available XML-based business process modelling languages for completeness using a set of meta-model concepts extracted from the candidate specifications. These concepts include the ability of the specification to include statistical data as part of the business process meta-model. The inclusion of statistical data for activities provides a business process monitoring framework with information to create relevant charts and alerts for its business users. The study shows that the Workflow Management Coalition's (WfMC) XML Process Definition Language is the only standardized interchange format that includes process statistics. However, data captured is general and caters mainly for simulation activities using statistical data such as costs and durations of tasks.

8 CONCLUSION

Fragmented collaborative and enterprise process visibility requires a framework that integrates processes, events and business information. This paper described a common event-based model for integrating disparate system processes and introduced the iWISE BAM software architecture for process performance monitoring. This approach brings together two important elements of enterprise modelling to provide a more cohesive view of processes, the events generated by processes, and the business data manipulated by the process. The iWISE framework allows users to configure and manage event-based process models using the iWISE Process Capture Tool and deploy these models to the iWISE Event Server. iWISE Legacy Listeners listen for system events and forward these to the Event Server. Continuous data integration is achieved by packaging business data with raw system events and sending it to the Event Server. The Event Server processes a continuous stream of event information and the iWISE Process Dashboard presents users with up-to-date process information. This generic approach is applicable to processes from various business domains. A health-care process and manufacturing process were presented in this paper which the iWISE framework can support to enhance business operational effectiveness through punctual process visibility.

REFERENCES

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

Agrawal, R., Gunopulos, D., and Leymann, F. (1998). Mining Process Models from Workflow Logs. In *Proceedings of the 6th International Conference on Extending Database Technology*, pages 469–483. Springer-Verlag.

Bobrik, R., Reichert, M., and Bauer, T. (2005). Requirements for the Visualization of System-Spanning Business Processes. In *The 16th International Workshop on Databases and Expert Systems Application*, Copenhagen, Denmark.

Casati, F. (2005). Industry Trends in Business Process Management: Getting Ready for Prime Time. In *The 16th International Workshop on Databases and Expert Systems Application*, Copenhagen, Denmark.

Casati, F. and Discenza, A. (2000). Modeling and Managing Interactions Amoung Business Processes. Technical Report HPL-2000-159, Hewlett Packard, HP Laboratories Palo Alto, CA.

Caulcutt, R. (2001). Why is Six Sigma so successful? *Journal of Applied Statistics*, 28(3-4):301–306.

Costello, C., Molloy, O., Lyons, G., and Duggan, J. (2005). Using Event-based Process Modelling to Support Six Sigma Quality. In *The 16th International Workshop on Databases and Expert Systems Application*, Copenhagen, Denmark.

Eder, J., Olivotto, G. E., and Gruber, W. (2002). A Data Warehouse for Workflow Logs. In *EDCIS '02: Proceedings of the First International Conference on Engineering and Deployment of Cooperative Information Systems*, pages 1–15. Springer-Verlag.

Eriksson, H.-E. and Penker, M. (2000). *Business Modeling with UML, Business Patterns at Work*. OMG Press.

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*, pages 27–29.

McGregor, C. and Schiefer, J. (2004). A Web-Service based framework for analyzing and measuring busness performance. *Information Systems and e-Business Management*, 2(1):89–110.

Mendling, J., Neumann, G., and Nuttgens, M. (2004). A Comparison of XML Interchange Formats for Business Process Modelling. In *EMISA: Information Systems in E-Business and E-Government*, Luxembourg.

Mitchell, B. (1992). The Six Sigma Appeal. *Engineering Management Journal*, 2(1):41–47.

Smith, B. (1993). Making war on defects: Six-sigma design. *IEEE Spectrum*, pages 43–47.

Stefanov, V. and List, B. (2005). A Performance Measurement Perspective for Event-driven Process Chains. In *The 16th International Workshop on Databases and Expert Systems Application*, Copenhagen, Denmark.

van der Aalst, W. and Weijters, A. (2004). Process Mining: A Research Agenda. *Computers in Industry*, 53(3):231–244.

White, S. (2004). Business Process Modeling Notation (BPMN), Version 1.0. Technical report, The Business Process Management Initiative (BPMI.org).