A shareable web service-based intelligent decision support system for on-demand business process management

Carolyn McGregor*

Health Informatics Research,
School of Computing and IT,
University of Western Sydney,
Locked Bag 1797,
Penrith South DC, NSW 1797, Australia
E-mail: c.mcgregor@uws.edu.au
*Corresponding author

Josef Schiefer

IBM Watson Research Center,
19 Skyline Drive, Hawthorne, NY 10606, USA
E-mail: josef.schiefer@us.ibm.com

Michael zur Muehlen

Wesley J. Howe School of Technology Management,
Stevens Institute of Technology,
Castle Point on the Hudson,
Hoboken, NJ 07030, USA
E-mail: mzurmuehlen@stevens-tech.edu

Abstract: The monitoring of business processes for performance measurement is challenged by the variety of inter and intra organisational units and information systems involved in the execution of these processes. In this paper we present a shareable, web service-based Intelligent Decision Support System (IDSS) for on-demand business process management, that we call the Solution Manager Service (SMS). The SMS allows organisations to outsource the collection, accumulation and transformation of information about their business processes from multiple distributed systems across multiple organisations in a centralised repository and share them among authorised parties, such as supply chain partners, clients or government agencies.

Keywords: business process management; monitoring; controlling; web services; workflow; Business Process Execution Language for Web Services (BPEL4WS).

Reference to this paper should be made as follows: McGregor, C., Schiefer, J. and zur Muehlen, M. (2006) 'A shareable web service-based intelligent decision support system for on-demand business process management', *Int. J. Business Process Integration and Management*, Vol. 1, No. 3, pp.156–174.

Biographical notes: Carolyn McGregor is the Head of Health Informatics Research (HIR) in the School of Computing and IT, University of Western Sydney. She received a PhD in Computing Science, where she developed new ways to use business strategy, business process definition and business enactment for automated business performance measurement. She is currently researching the application of this research within the domain of health and medicine. Her research interests include intelligent decision support systems, business process performance management, multi-agent data mining, data stream event correlation, web services, patient journey modelling and health informatics.

Josef Schiefer received a PhD in Information Systems from the University of Vienna and was a Research Assistant in the Department of Software Technology at the Vienna University of Technology. He joined IBM's Thomas J. Watson Research Center where he continued his research in the areas of business process management, business process intelligence, workflow management and data warehousing. Currently, he is the Technical Director at Senactive Inc., a company offering software for real-time sense and respond solutions.

Michael zur Muehlen is a Renowned Expert in the field of process automation and workflow management. Currently, he is an Assistant Professor of Information Systems at Stevens Institute of Technology in Hoboken, NJ. Previously, he was a Senior Lecturer in the Department of Information Systems, University of Muenster, Germany and also a Visiting

Lecturer at the University of Tartu, Estonia. He received a PhD and an MSc in Information Systems from the University of Muenster, Germany. His research interests include management aspects of workflow technology, controlling of business processes, resource management and inter-organisational process management.

---

1 Introduction and motivation

The continuous supervision of business process performance is of increasing importance to process designers, system architects and business users. Continuous improvement has been advocated in the management literature for more than 50 years, for example, in the Deming cycle, which is used as a template for total quality management and six sigma efforts (Deming, 1992). While the interest in process-oriented organisational structures has peaked in the re-engineering wave of the early 1990s, the continuous realisation of reorganisation benefits is at the focus of corporate strategy today (Hammer, 2001a,b). Reaping the benefits of a process-oriented organisation requires the continuous maintenance and control of the organisation's business processes.

Business Process Management consists of the planning, implementation, enactment and controlling of processes, forming a life cycle that leads to continuous process improvement. It enables companies to stay adaptable to environmental and internal changes as well as realise efficiency gains through exploiting cost-effective ways to produce goods and perform services (zur Muehlen, 2003).

Separated from operational systems, many organisations built data warehouse and business intelligence applications for strategic planning and decision-making. It has become apparent that the information and analysis methods they provide are also vital for tactical day-to-day decision making processes and many organisations can no longer operate their businesses effectively without them. Consequently, there is a trend towards integrating decision processing into business processes in an organisation. Furthermore, a new driving force in business strategy towards outsourcing and inter-organisational business processes is emerging and moving companies far beyond the confines of the individual organisation.

On the one hand, business process management helps to identify efficiency constraints and organisational bottlenecks faster than traditional reporting infrastructures. On the other hand, regulatory bodies require managers to implement controls for their organisation's processes in order to audit their accuracy. The latter has been highlighted by the Sarbanes-Oxley Act of 2002, which requires that:

> "the signing officers [..] are responsible for establishing and maintaining internal controls" (sarbanes-Oxley Act of 2002, 2002).

Collecting information about an organisation's business processes is complicated by the lack of a common infrastructure. Frequently, business processes cross organisational boundaries and they often involve a variety of information systems, frequently with disparate data repositories.

For the execution and monitoring of business processes, many organisations are increasingly using Workflow Management Systems (WFMS) to improve the efficiency of their processes and reduce costs. During the execution of the business process, WFMS record many types of events, such as the start and completion time of each activity, the assigned resources and the outcome of the execution. Major WFMS lack capabilities for transforming, accumulating and condensing audit trails of distributed business processes and using this information for monitoring and analysis purposes to provide feedback about the performance of business processes (in particular inter-organisational business processes).

Leymann and Roller point out that audit trail data from automated business processes can quickly increase to a sizable amount (Leymann and Roller, 2000). If several organisations are involved, for example, if processes across an integrated supply chain are to be supervised, the volume of information can quickly reach levels that negatively impact the operational performance of enterprise applications. A way to handle data intensive operations is the outsourcing of data processing to external parties. An example for this is the upspring of off-site service providers filtering incoming e-mail traffic for unsolicited messages and potential viruses (Kontzer, 2003). The outsourcing of process management services allows for the sharing of process information across multiple organisations. The positive economic effect of shared information in an integrated supply chain has been demonstrated in a number of studies (Cachon and Fisher, 1999; Gavirneni et al., 1998; Holten et al., 2002; Lee et al., 1999).

The proposed infrastructure is a follow on from several independent research studies conducted by the authors of this paper. Jeng and Schiefer have developed an agent-based architecture with the aim of providing continuous, real-time analytics for business processes (Jeng and Schiefer, 2003). For the analytical processing they introduce an agent framework, that is, able to detect situations and exceptions in a business environment, perform complex analytical tasks and reflect on the gap between current situations and desired management goals. MCGregor and Kumaran have presented a Solution Management framework that analyses workflow audit logs, utilising decision support system principles and agent technologies to feedback performance measures (McGregor and Kumaran, 2002a,b). This framework forms part of the Intelligent Workflow Monitoring System (IW-MONS) meta-methodology (McGregor, 2002a,b; McGregor and Edwards, 2000). An extension for this framework using web services was proposed by McGregor and Schiefer, who state that current web service frameworks do not include the functionality required for web service execution performance measurement from an organisational perspective (McGregor and Schiefer, 2003, 2004). Part of this work is the extension of the Business Process Execution Language for Web Services (BPEL4WS) with mechanisms to obtain performance information (Andrews et al., 2003). zur Muehlen has developed a generic architecture for workflow-based Process Information Systems (zur Muehlen, 2000, 2001, 2003). Based on the flow of management information in an organisation, he developed a cybernetic model of three distinct feedback levels for automated process regulation, operational process management and strategic process management. This model and the associated process-oriented data warehouse have been implemented in a prototype based on the workflow management system Carnot. This research paper utilises this previous research by enhancing and linking individual components to create the shareable, web service-based Intelligent Decision Support System (IDSS) infrastructure for on-demand business process management, that we call the Solution Manager Service (SMS).

The rest of this paper is structured as follows. Related work is first reviewed and the proposed infrastructure is positioned among existing process management products. The functional requirements for a business process management infrastructure are then defined. The architecture of the SMS is then introduced. The internal components of the SMS are then defined in more detail. The use of the SMS in different scenarios is then described. This paper ends with an outlook on future work.

2 Related work

The use of workflow audit trail information for controlling purposes has been recognised by the scientific community only recently. In 1996, McLellan (1996) provided the first survey of the analytical opportunities arising from audit trail information. He discusses the analysis of historical process data through the evaluation of audit trail data in terms of workflow metrics providing mainly statistical evaluations as well as the run-time detection of late cases and overdue tasks.

The Workflow Management Coalition (WfMC) defined Interface 5 for workflow audit data within their generic workflow reference model (Hollingsworth, 1995) that specifies the elementary data structures a workflow management system should record about the execution of workflow instances (WfMC, 1999). The standard consists of an audit trail abstract data format and a list of standard events that should be recorded. The evaluation of this information is not addressed in the standard. A proposal for an API has been submitted to the WfMC (1998) but no progress has been made on this standard since 1999.

The design of a process analysis tool, WorkFlow Analyser, was presented by Derszteler (2000) as part of his PhD. A partially functional prototype was implemented in Forest&Trees using audit trail data from Siemens Nixdorf's workflow management system WorkParty and to-be data from the business process modelling tools ARIS and Bonapart. While it provided several quantitative evaluation methods, the platform-specific implementation and reliance on a single (and discontinued) workflow management system limit the general applicability of his approach.

A more process-specific prototype was developed by Raufer, who discussed the controlling of workflow-based processes by focusing on a specific process from a manufacturing domain case study based on the workflow management system COI (Raufer, 1997). This specific process was enhanced with cost information as well as target work- and cycle-times. The system architecture cannot be generalised easily. Weiss (Weiß, 1998, 1999; Weiß and Zerbe, 1995) presented a similar prototype of a workflow-driven activity-based costing system for the workflow management system Staffware that was focused on a single evaluation method.

The COPPA project (Computer-based Process Performance Measurement) conducted at the University of Fribourg, Switzerland, dealt with the design of a performance measurement system (Kueng, 1998, 2000, Kueng and Krahn, 1999). Their three-stage approach was:

1 survey the market and corporate practice of performance measurement

2 outline the architectural and functional requirements of a performance measurement system and

3 implement a prototype of the performance measurement system.

While their research positioned process performance measurement at a higher level of abstraction, the SMS also incorporates the analysis of operational data.

Three process controlling system prototypes were implemented in the CONGO (Controlling and Monitoring of distributed Workflows for continuous Process Improvement) project, (zur Muehlen, 2000, zur Muehlen and Rosemann, 2000). The first prototype (PISA I) was a feasibility study based on a Microsoft Access database that was designed to analyse information from the ARIS Toolset and IBM FlowMark. It implemented elementary cube based evaluation methods with process, abstraction and organisation dimensions. PISA II (also based on Microsoft Access) used more sophisticated evaluation methods, such as the hedonic wage model allowing additional evaluations on process objects, such as a cluster analysis. PISA III, implemented in Java, is a fully distributed system architecture with database and client independence. The stand-alone PISA Server coordinates the PISA clients and delivers the evaluation methods, graphical representation panels as well as the results of internal evaluations on the audit data. The PISA Client is a Java applet executable with a Java-capable web-browser. Data source adapters enable the system to integrate the audit trails of different workflow systems by mapping the contents of a workflow audit trail database with the audit trail repository of the PISA system. While portable, the PISA system does not support more recent developments such as web services or XML.

The IDS Process Performance Manager is a commercial performance analysis tool that mainly integrates source data from modules of the SAP R/3 ERP system into a relational database and calculates predefined ratios (IDS Scheer AG, 2000). These ratios are frequency and time-related and computed at different aggregation levels, reflecting recipients at the operative level, middle management and executives. Staffware offers the Process Performance Manager as a customised OEM version for their workflow management system.

Casati et al. at Hewlett Packard Laboratories have developed a Process Data Warehouse that collects metrics from the workflow management system HP Chang*e*ngine (Bonifati et al., 2001; Casati et al., 2002; Sayal et al., 2002). They use process mining algorithms to derive patterns from workflow audit trail data. The Business Process Cockpit uses multi-coloured bar graphs to illustrate time-based process attributes, such as the activity processing times.

The previously discussed commercial solutions use proprietary process models, which make them difficult to use in a wide range of contexts. They mainly focus on intra-organisational business processes and lack the ability to share the monitoring infrastructure among business partners.

Lambros et al. (2001) state the interactions between the service registry, service requestor and service provider to establish and commence a web service relationship. While this model allows for the establishment and enactment of the relationship, it provides no mechanism to monitor this relationship for performance measurement by any of the participating organisations.

The SMS infrastructure allows organisations to collect information about their business processes in a centralised repository, and share them among authorised parties, such as supply chain partners, clients or government agencies. A key contribution of this research is that the interaction with this IDSS, is via a set of web services. In addition, the infrastructure boasts several innovative agent based modules that support different components, providing the intelligence within this IDSS. The data management layer, proposes data formats for shareable, on-demand business process management. Data-driven security policies maintain appropriate data access.

This infrastructure provides many benefits. It accumulates and condenses audit trail information from multiple distributed systems across multiple organisations within one environment. It is an on-demand outsourced approach freeing the organisation from internally processing audit trails and event logs. The processing of events is in near real time making the latest performance information available much earlier than with traditional Extraction, Transformation and Load-(ETL)-based approaches.

Therefore, the Solution Manager uses a different approach for the data integration than used in existing research approaches such as COPPA and PISA.

3 Requirements for a business process management infrastructure

Dunn (1990) distinguishes four quality criteria for software: reliability, usability, maintainability and adaptability. Kazman et al. (1994) use a different schema to classify the qualitative properties of software:

- *quality system output*: this includes correctness, security, reliability and availability
- quality aspects for system developers and administrators, such as maintainability, portability, adaptability and scalability
- quality system use, for example, ease of use, predictability and learnability.

Since we present an infrastructure that provides data structures as a result rather than a specific user interface, the latter criteria apply only to a limited extent. The goal of this research is the development of an infrastructure that achieves a high-quality system output by being: business process focused, reliable, available on-demand, shareable, responsive, proactive, reactive and with access to near real-time data. In order to realise the system developers' quality criteria, we focus on a standards-based infrastructure.

The goals of the SMS infrastructure are given as follows:

Business process focused: while traditional business intelligence solutions focus on strategic decision support, deployment of data warehouse solutions for operational decision-making that are seamlessly integrated with the business process and provide feedback about the business performance to improve the speed and effectiveness of business processes are becoming increasingly important. The SMS enables operational decision-making, that is, seamlessly integrated with the business process and provides feedback about the business performance through the Performance Monitor and the Analytical Processor.

Reliable: as organisations are required to maintain performance information in order to satisfy legal auditing requirements, the infrastructure must be reliable. The SMS infrastructure is external to the organisations providing them with an independent infrastructure to satisfy legal auditing requirements, thus improving the reliability for the individual organisation.

On demand: Foster (2003), states that computer grids assemble computers so that users can call on resources as required for processing, storage, data and software, regardless of location, from any suitable supplier. Supplying services on demand enables the user to request and use e-business services only as and when they need it. The SMS architecture enables the outsourcing of the business process management infrastructure, therefore, enabling the user to request and use business process management only as and when they need it.

Shareable: as organisations can be both the supplier of web services and the user of web services supplied by others, the web services of interest extend beyond the bounds of their organisation. They require the ability to measure the performance of all web services used by that organisation. Ideally the information relating to the web service enactment should be stored once and accessible to all appropriate parties based on appropriate security access controls. The business process management information is stored once within the data management layer of the SMS and accessible to all appropriate parties based on appropriate security access controls.

Responsive: organisation's today operate in a highly volatile environment constantly impacted by internal and external change and require a performance management infrastructure that is able to react and adjust to these changes. The SMS infrastructure is data driven and highly responsive in its ability to adjust to the changing needs of the organisations it supports.

Proactive and reactive: proactive monitoring provides information on how the business is changing and on the future direction of management. Reactive monitoring provides information on how the organisation has performed such as what business commitments have been violated; and often there is a lag time. Depending on the information gathered, the information can be used to modify the business process to fix or respond to what has already happened or can provide feedback for management changes to prevent problems in the future. Proactive and reactive monitoring within the SMS is provided by the Performance Monitor and the Analytical Processor.

Near real time: traditional data warehouse solutions, report generators, OLAP and data mining tools typically do not allow monitoring of business processes on a continuous basis. Existing ETL tools achieve high efficiency in loading large amounts of data periodically into the data warehouse system. The new desire for monitoring information about business processes in near real-time is breaking the long-standing rule that data in a data warehouse is static except during the downtime for data loading (McGregor and Schiefer, 2003). The Event Stream Processor (ESP) within the SMS provides a robust, scalable data staging environment that supports continuous integration of a large number of near real-time events. It handles each event with a lightweight Java thread, rather than a heavyweight operating system process enabling the monitoring information about business processes in near real-time.

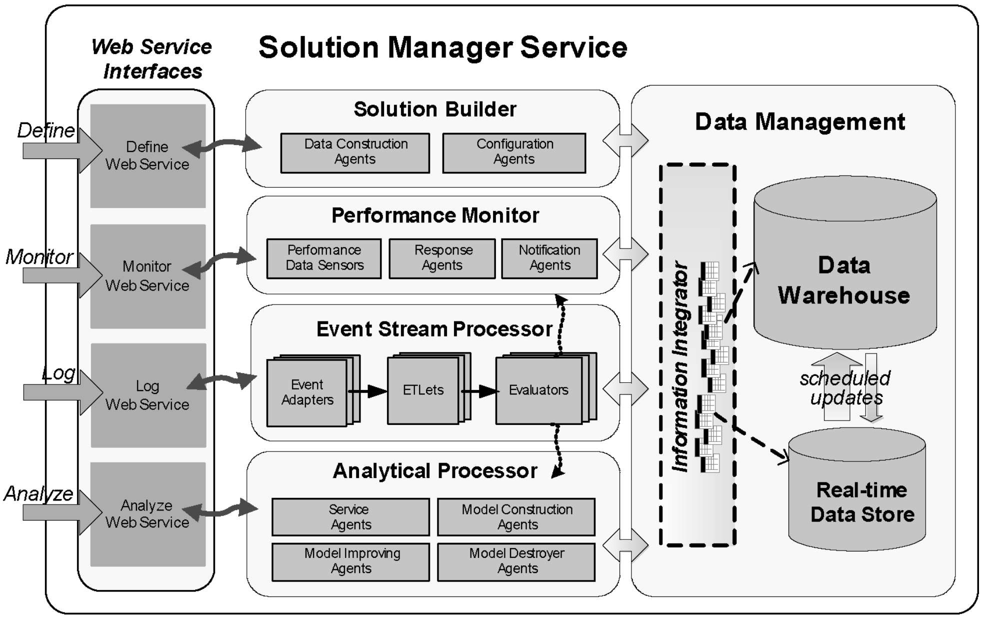
Standards based: organisations moving to utilise web services require an infrastructure to recapture the organisation's performance knowledge that is well aligned architecturally with web service principles. In addition, to minimise the data transformation steps, the data formats should be a standardised format such as XML. The interaction between organisations and the SMS is via a set of web services, enabling this infrastructure to be based on well accepted standards.

4 SMS infrastructure

The SMS infrastructure is shown in Figure 1 and includes five major components, with the first four accessible by web services:

- *Solution builder* is a build-time component that captures metadata and performance objectives for business processes to be monitored. It uses this data to set up and initialise the runtime components and the data management layer.
- *Performance monitor* is a run-time component enabling organisations to define and change service-level agreements based on selected performance measures.
- *ESP* provides a scalable data staging environment to continuously integrate and transform events buried in the audit trails of business processes, during run-time.
- *Analytical processor* provides a run-time interface for retrieving analysis data from the data warehouse or real-time data store and further manages the analytical processing models.

Figure 1 Solution Manager – web service architecture



Data management provides persistent storage of business process metadata, performance objectives established during build time and event data and performance metrics about business processes instances received during run-time.

In the following sections we will discuss the architectural components in detail.

5 Data management

Several data structures are used in the SMS. On the event log side, a unified format for incoming events is required. Internally, the SMS contains a data store to make the incoming events persistent and to allow evaluations of events over time. Finally, the monitor web service and the analysis web service rely on their own XML formats to deliver aggregated information about events or higher-level analytics. In this section, we discuss the individual data formats described above.

5.1 A data model for business process audit trail information

Since the SMS records events that are generated during the enactment of business processes, a generic event structure is required that is capable of accommodating different process modelling paradigms such as activity networks (Leymann and Altenhuber, 1994). Petri-nets (Salimifard and Wright, 2001; van der Aalst and van Hee, 2002) or statecharts (Harel, 1988). Figure 2 shows a reference data model for audit trail information using the entity-relationship diagram notation (extended from (zur Muehlen, 2003)).

We treat a business process as a collection of discrete activities, which may be executed by either a technical resource or a human performer. These concepts are generalised to the entity *process object*. A process object can be process or activity (at the instance level) and is associated with a state at any given time. State changes are represented as event types (e.g. a process instance moves to state *completed*). The ternary relationship between *event type*, *time* and *originator* (representing the system component or the human user causing the event) gives us the notion of *event*, which may be associated with a process object. The optional link to the process object provides access to the process that the event originated from. Since process objects can be associated with business objects (such as data fields in operative information systems or more complex structures such as customer records), it is possible to navigate from a single event to the associated business data. For human-generated events, the originatorID provides a link to the organisational model of the originating domain.

This data structure leads to the development of an XML schema for business process events, which is depicted in Figure 3. Each XML event message contains the timestamp when the event occurred, the ID of the event originator and the type of the event. Optional elements of the message include further event details, which are represented as name-value pairs and details about the process object. The process object contains at least the ID of the process object instance in question and the ID of the model this instance was derived from. To support hierarchical processes, a message may contain information about the root process object that serves as the container for the process object, which caused the event. Additionally, an identifier of an associated business

Figure 2 Data model for audit trail information

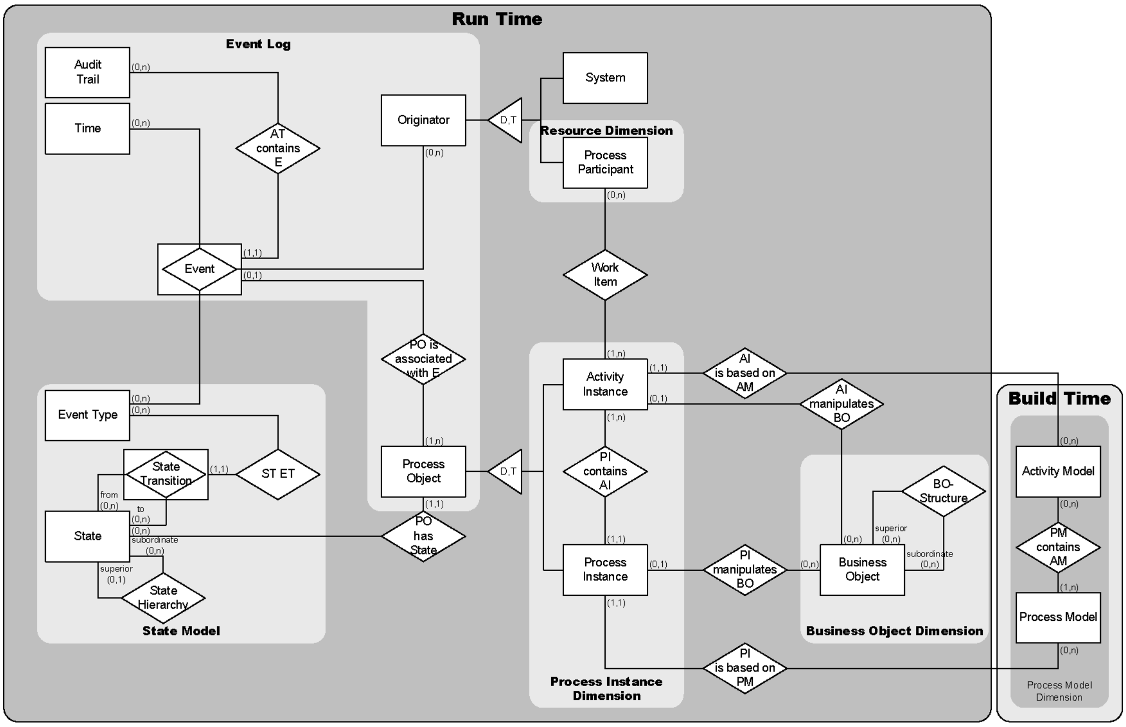
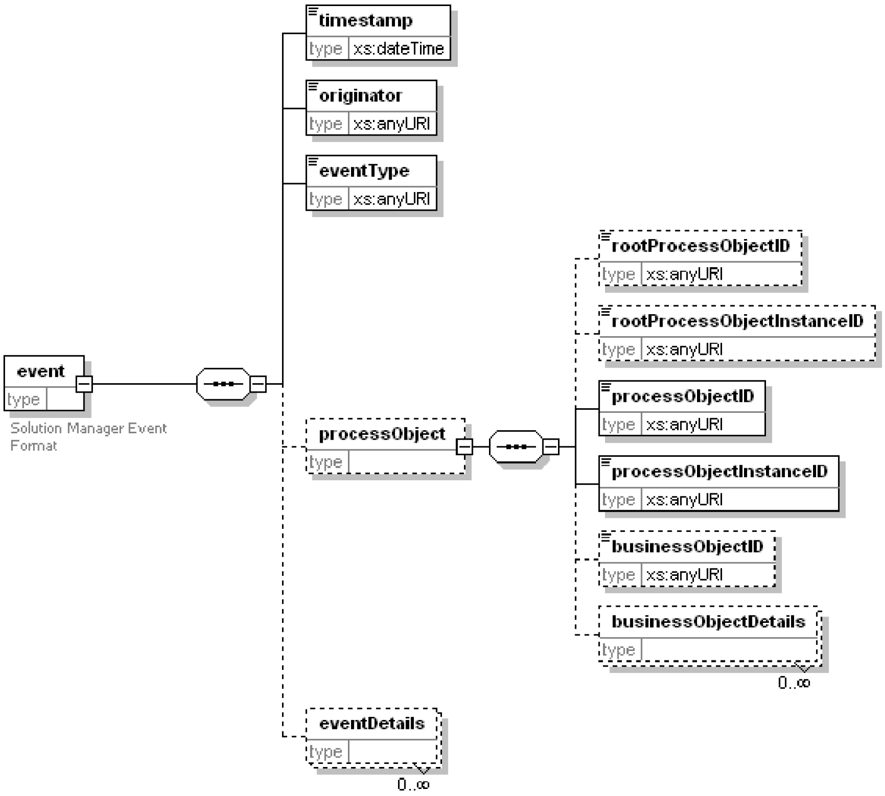


Figure 3 XML schema of business process events



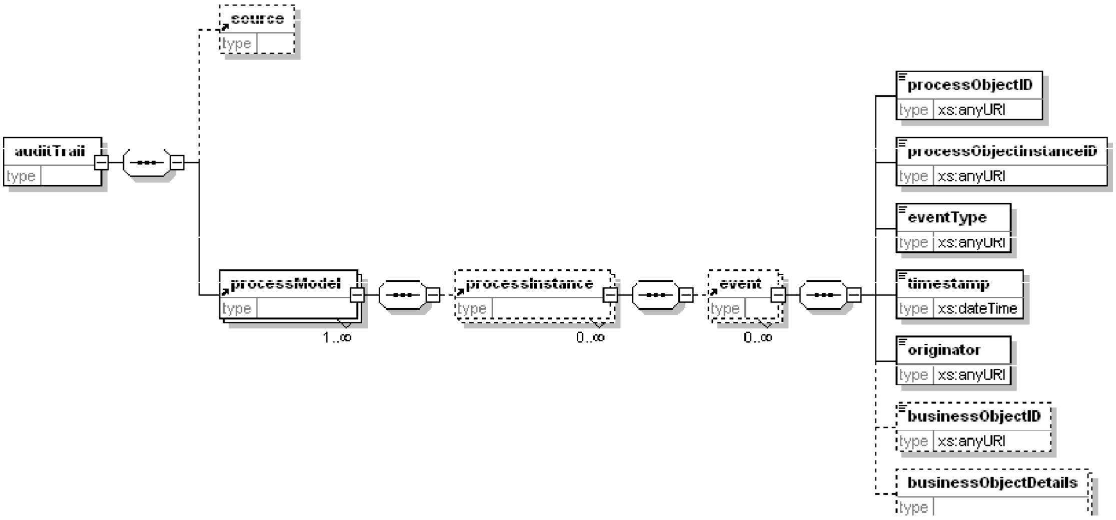
object and additional information about this object in the form of name-value pairs may be included in the message.

5.2 A data storage structure for business process events

Once process events have been received by the SMS, they are manipulated by the ESP, which is detailed in the Log Web Service and ESP section. The results of these transformations can be the storage of the event in a data warehouse structure for future evaluation. In addition, the event may be temporarily stored in a real-time data store to support monitoring operations. The storage requirements for these two repositories are different, since ex post analytics typically rely on an aggregation of atomic process events, while monitoring applications are focused on the current state of a process instance or particular events that signal an exception in a process.

Events are collected in the real-time data store according to their root process ID. Through a simple XSL transformation, the atomic event schema is converted into a structure that is rooted in the audit trail of a particular process instance. This schema is derived from the workflow mining format proposed by van der Aalst and van Dongen (2002), that is, it supports the automated discovery of process structures with tools such as EMiT (described in van der Aalst and van Dongen, 2002), in case the original model of the business process is not available (Figure 4).

Figure 4 Audit trail format for real-time data store



Information stored in the real-time data store is transferred into the data warehouse on a regular basis. The data warehouse serves as the central repository for high-level analytics, and is built around star schemas. In order to facilitate a multitude of evaluations, different evaluation dimensions need to be supported by this data warehouse. Separate data marts focus on individual evaluation dimensions. The evaluation dimensions and facts within the fact table are data driven based on the information received by the *define web service* and built by configuration agents within the solution builder. This enables the data warehouse structure to be flexible to support changing needs over time. An example for such a data mart is shown below.

The content of audit trail data is characterised by its detailed notion of time, which is one identifying element of the triple relationship *time-process object-event type*, which defines the core of the business process audit trail. Since the audit trail provides timestamps, but not aggregate information about time periods, it is useful to create a time-oriented data mart from raw audit trail data. This data mart contains a central fact table, which contains (calculated) information about the activity instances. Figure 5 shows the data structure for a time-oriented data mart. The central fact table contains references to the activity model that the activity instance was derived from, the process instance that formed the context for the activity instance, the resource (i.e. the process performer) that executed the activity, and the business object that was manipulated in the context of the activity. The facts contained in the fact table are the calculated values for the different processing time categories.

The references to activities, processes, resources and business objects opens access to four evaluation dimensions, which can be used to formulate queries. For example, the business object perspective may be used to determine those activity instances with the longest idle time for a specific business object type.

6 Define web service and solution builder

6.1 Define web service

The *define web service* provides services to describe a web service from a performance measurement perspective. In addition, the organisation may establish measures that relate to individual partner relationships. This functionality is different to that offered by the service registry as this records business level information relating to the service as opposed to the technical level information relating to the service published in the service registry.

A high-level sequence diagram for the registration and definition of web services is shown in Figure 6. The organisation (shown as the Service Supplier) has established a web service to enable its customers to place orders and has published that web service within the service registry. In publishing the web service, they provided a service description containing the details of its data types, operations, binding information and network location. This definition was constructed in WSDL. In addition, this organisation is required to define its web service to the SMS using the *define web service* that has been registered with the service registry previously.

Figure 5 Data mart with time focus

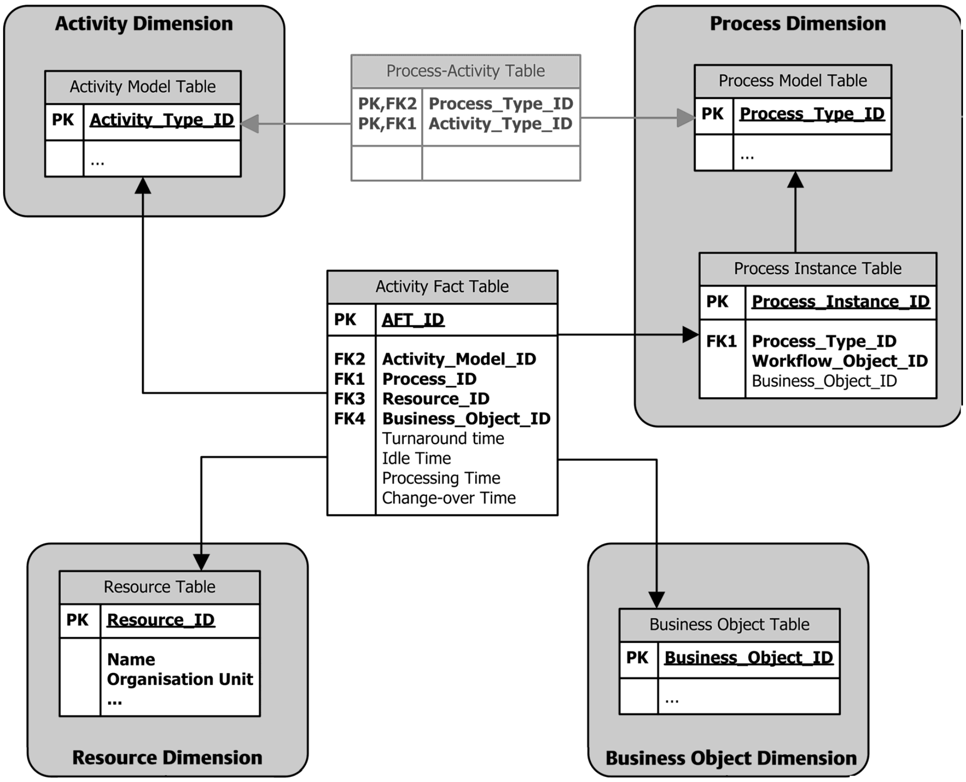
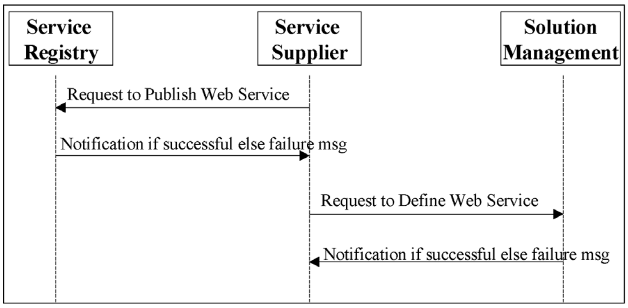


Figure 6 Web service registration and definition



The SMS *define web service* supports two separate functions (McGregor and Schiefer, 2003, 2004). Firstly, it provides the environment to define general performance measures for the web service. Secondly, it provides the environment to define performance measures for a partner or partner group that participate in the web service. The *define web service* details given in Figure 7 are an example of fields used when it is used to define the web service in general.

Figure 7 Define web service (general)

```
Name:   DefineWebService
Input:  WebServiceDefinition
            WebServiceDescription
                Web Service ID
                Description
            Owner
                OwnerID
                Name
                Address
            WebServiceStates
                StateList
                StateTransitionRules
            Product
                ProductGroup
                Product Description
            ServicePriceDetails
                ServicePrice
                SPStartDate
                SPEndDate
            CostDetails
                Cost
                CStartDate
                CEndDate
            Measures
              ProcessTime
                        - ProcessTimeID
                        - ServiceLevel
                        - ServicePercentage
                        - PTStartDate
                        - PTEndDate
              Throughput
                        - ThroughputID
                        - Target
                        - Frequency
                        - TStartDate
                        - TEndDate
              Rework
                        - ReworkID
                        - Target
                        - RStartDate
                        - REndDate
Output:
        WebServiceDefinitionResponse
            Web Service ID
            ResultNotification (Boolean)
```

Source: Lee et al. (1999).

When establishing the general definition for a web service or BPEL4WS process, the definition contains the name and owner of the web service, a list of the valid states for this web service that are to be tracked as milestones for the process and the associated transition rules for the valid state transitions. The product group that is applicable for this process can also be listed. It may also contain measures relating to that web service, for example: the web service cost, price, processing time service level and or rework targets. Each measure has a start and end date to enable multiple values for each measure over time to be recorded where required. The *define web service* will return a flag confirming that the definition has been successful.

By default, the security policy that is established for the owner of the web service provides access to review all information stored in relation to instances of their web service.

The *define web service* can also then be used to additionally define measures that specifically relate to a partner or partner group. Additional input fields are required by the *define web service* as given in Figure 8 that define the partner for which these measures relate. Partner-based web service definitions can be established for individual partners (where the individual partner is defined as the PartnerGroup) or groups of partners. Partner groups can be defined so as to construct partner group hierarchies and partners can belong to multiple partner groups if required. Where a measure is defined for a partner or partner group, this overrides any values defined in the general definition in relation to that partner or partner group only. The security policy for the partner or partner group is also contained within this definition. By default, partners may only access web service instance information for web service instances that they have participated in. The owner of the web service may choose:

- the level of access given to the partner or partner group (all or summary)
- the time window of information available to the partner or partner group (all, week, month, quarter and year) and
- a list of the performance measures available to the partner or partner group that corresponds to the performance measures as defined within the measures section of the definition.

6.2 *Solution builder*

The solution builder is an automated agent-based build-time configuration module that is invoked when the *define web service* is used. Data constructor agents within the solution builder capture the metadata and performance objectives for the business processes to be monitored and load the information into the activity model and process model tables within the data warehouse and run-time data store to set up and initialise the data management layer. The data stored within the definition tables will be used later by the *analyse web service* for security policy verification and to combine with actual instance data in order to return performance results to the requestor. Configuration agents establish the star schema structures within the data warehouse based on the information received by the *define web service*. In addition, configuration agents establish rule sets for the model constructor, improver and destroyer agents within the analytical processor that will create models based on web service instance analysis applicable for the web service that has been defined.

Figure 8 Define web service – partner group

```
Name:   DefineWebService
Input:  WebServiceDefinition
                WebServiceDescription
                        Web Service ID
                Owner
                        OwnerID
                Product
                        ProductGroup
                ServicePriceDetails
                        ServicePrice
                        SPStartDate
                        SPEndDate
                CostDetails
                        Cost
                        CStartDate
                        CEndDate
                PartnerGroup
                        PartnerGroupID
                        Name
                        Description
                Security Policy
                        AccessLevel
                        AccessTimeWindow
                        AccessMeasures
                                - AccessMeasureList
                Measures
                    ProcessingTime
                                - ProcessingTimeID
                                - ServiceLevel
                                - ServicePercent
                                - PTStartDate
                                - PTEndDate
                    Throughput
                                - ThroughputID
                                - Target
                                - Frequency
                                - TStartDate
                                - TEndDate
                    Rework
                                - ReworkID
                                - Target
                                - RStartDate
                                - REndDate
Output:
        WebServiceDefinitionResponse
                ResultNotification (Boolean)
```

6.3 *Approaches for defining web services*

The *define web service* interface to solution builder provides organisations with flexibility in choosing how to interact with the *define web service*. Two examples are detailed below to illustrate this flexibility, namely using BPEL4WS (Andrews et al., 2003) and using the WfMC XPDL Interface 1 (WfMC, 2002).

6.4 *BPEL4WS*

To enable a business process, defined using BPEL4WS, to contain the SMS *define web service* information an additional element is required (McGregor, 2003). This additional element is added by utilising the BPEL4WS extensibility rules (see Section 6.3 of Andrews et al, 2003). The extension namespace is defined as:

<process name= 'Process Order'

xmlns:sm='http://www.solutionmanager.com/wsdl/define-process'/>

<extension namespace=http://www.solutionmanager.com/wsdl/define-process/>

Figure 9 demonstrates how the *define web service* details would be represented within a BPEL4WS process element.

Figure 9 BPEL4WS process additional element

```
<sm:WebServiceDefinition>
  <sm:WebServiceDescription>
   <sm:WebServiceID>Order Process </sm:WebServiceID>
   <sm:Description>Process for Orders </sm:Description>
  </sm:WebServiceDescription>
  <sm:Owner>
   <sm:OwnerID>LP</sm:OwnerID>
     <sm:Name>The Supply Company</sm:Name>
  </sm:Owner>
  <sm:WebServiceStates>
   <sm:StateList>
   <sm:State>Receive Order</sm:State>
     <sm:State>Process Order</sm:State>
     <sm:State>Ship Order </sm:State>
     <sm:State>Finalise Order </sm:State>
   </sm:StateList>
   <sm:StateTransitionRules>
     <sm:Transition>
       <sm:From>Receive Order</sm:From>
       <sm:To>Process Order </sm:To>
     </sm:Transition>
     <sm:Transition>
       <sm:From>Process Order</sm:From>
       <sm:To>Ship Order </sm:To>
     </sm:Transition>
     <sm:Transition>
       <sm:From>Ship Order</sm:From>
       <sm:To>Finalise Order</sm:To>
     </sm:Transition>
   </sm:StateTransitionRules>
  <sm:WebServiceStates>
  <sm:Product>
   <sm:ProductGroup>All</sm:ProductGroup>
   <sm:Product Description>All Products
   </sm:Product Description>
  </sm:Product>
```

Figure 9 BPEL4WS process additional element (continued)

```
   <sm:Measures>
         <sm:ProcessTime>
       <sm:ProcessTimeID>CT1</sm:ProcessTimeID>
       <sm:ServiceLevel>4</sm:ServiceLevel>
       <sm:ServicePercentage>0.9</sm:ServicePercentage>
       <sm:PTStartDate>01/01/2003</sm:PTStartDate>
       <sm:PTEndDate>31/12/2003</sm:PTEndDate>
         </sm:ProcessTime>
         <sm:Throughput>
       <sm:ThroughputID>T1</sm:ThroughputID>
       <sm:Target
>500</sm:Target>
       <sm:Frequency>weekly</sm:Frequency>
       <sm:TStartDate>01/01/2003</sm:TStartDate>
       <sm:TEndDate>31/12/2003</sm:TEndDate>
         </sm:Throughput>
         <sm:Rework>
       <sm:ReworkID>R1</sm:ReworkID>
       <sm:Target>0.1</sm:Target>
       <sm:RStartDate>01/01/2003</sm:RStartDate>
       <sm:REndDate>31/12/2003</sm:REndDate>
         </sm:Rework>
   </sm:Measures>
</sm:WebServiceDefinition>
```

Source: McGregor (2003).

6.5 WfMC XPDL interface 1

The metadata and performance objectives information for a business process may be defined and described using the WfMC XPDL interface 1 (WfMC, 2002). The XPDL XML document is processed by an XML Stylesheet (XSLT) to reformat the information into the required XML format for the *define web service*.

7 Log web service and ESP

The *log web service* provides the mechanism to gather the time-sequenced record of all status changes of a business process contained in an audit trail. BPEL4WS flows essentially implement a layer on top of WSDL, with WSDL defining the specific operations allowed and BPEL4WS defining how the operations can be sequenced. A BPEL4WS document leverages WSDL in three ways:

1. Every BPEL4WS process is exposed as a web service using WSDL. The WSDL describes the entry points for external services to interact with the process
2. WSDL data types are used within a BPEL4WS process to describe the information that passes between requests and
3. WSDL might be used to reference external services required by the process. BPEL4WS processes specify stateful interactions involving the exchange of messages between partners.

The state of a business process includes the messages that are exchanged as well as intermediate data used in business logic and in composing messages sent to partners. Variables provide the means for holding messages that constitute the state of a business process. The messages held are often those that have been received from partners or are to be sent to partners. Variables can also hold data that are needed for holding state information related to the process and never exchanged with partners. Variables identify the specific data exchanged in a message flow, which typically maps to a WSDL message type. When a BPEL4WS process receives a message, the appropriate variables are populated so that subsequent requests can access the data.

7.1 Approaches for auditing business processes

For auditing business processes, there are principally four options:

1. include the auditing mechanism as a partner within the BPEL4WS process
2. instrumentation of web service requests of the BPEL4WS process
3. utilising the auditing service of a workflow engine used for enacting the BPEL4WS process or
4. using probes in the operational systems that track state changes of the business process.

All four options have both the advantages and disadvantages and are discussed in Section 7.2.

7.2 Log web service

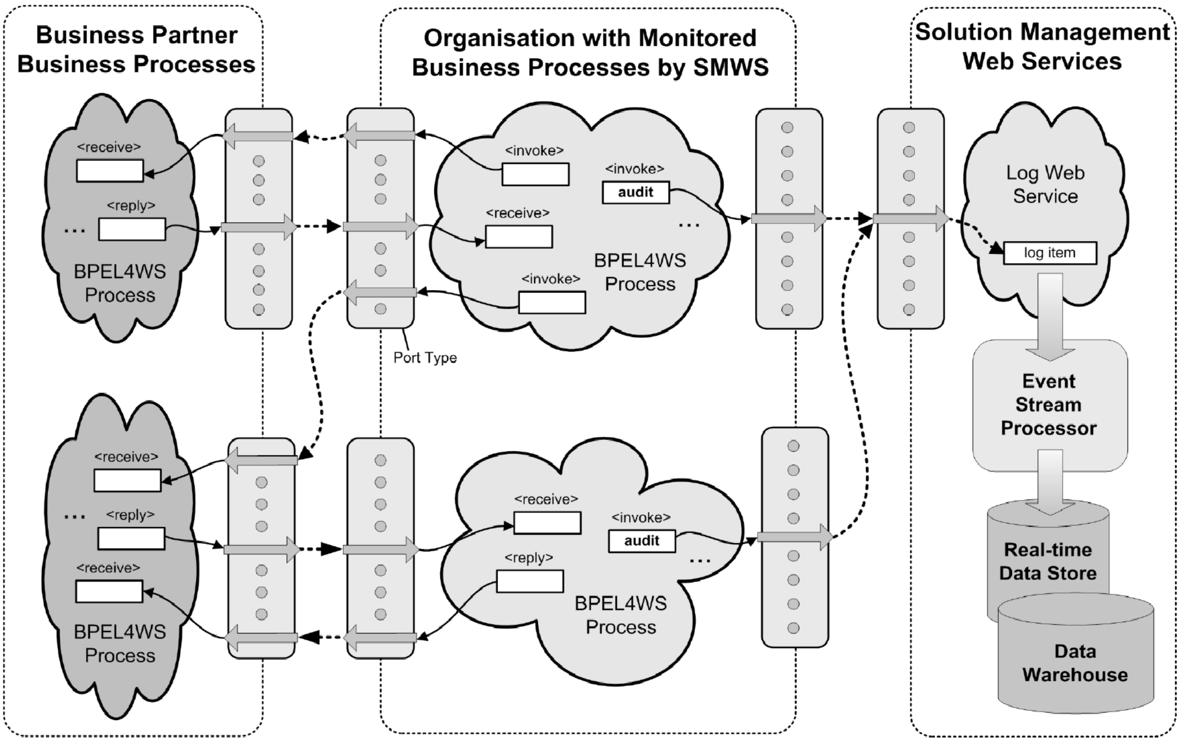
7.2.1 Auditing mechanism as a partner within the BPEL4WS process

McGregor (2003) defines a method to enable business processes defined using BPEL4WS to log audit information to a *log web service*, by establishing the Solution Management Service as a partner within the BPEL4WS process definition.

Figure 10 shows an example with BPEL4WS flows with embedded auditing operations that invoke the log web service of the Solution Management Service. Using that approach, a log request can be executed at any stage within the BPEL4WS process. Through the use of the <*assign*> activity construct values can be assigned to the message for holding the logging data (*webserviceLogItem*) and the inclusion of the following invoke request:

```
<invoke partner='SolutionManager'
        portType='sm:auditlogPT'
        operation='logitem'
        inputVariable='webserviceLogItem'
        outputVariable='logitemResult'>
</invoke>
```

Figure 10 Web service logging of BPEL4WS processes



The major advantage of this approach is the simple integration of audit trail information from various business partners. Multiple business partners can utilise the log web service for centrally capturing audit trail information about their BPEL4WS processes.

Although this approach enables seamless integration of the SMSs, the overhead of invoking the *log web service* for every state change can become a potential bottleneck. For instance, an order process with a large number of audits with the magnitude of thousands of process instances a day can result in potentially millions of log requests. For that reason, the direct invocation of the log web service within BPEL4WS processes might be not suitable for very critical or large-scale solutions and other mechanisms as detailed in the following options should be considered.

7.2.2 Inception of web service requests

Using this option, a web service gateway is used to intercept any web service request and extract the data needed for the auditing. The major advantage of this option is that the web service auditing is fairly straightforward and many tools are available that allow an interception of web service requests (e.g. IBM Web Service Gateway). However, the web service requests of BPEL4WS processes have a limited visibility to internal process information. For instance, the web service requests do not include details about the process instances or variables used by the workflow engine to control the process execution. Therefore, with this option only an incomplete audit trail can be extracted which often results in a more complex audit trail processing in order to regain the process information.

7.2.3 Auditing services of WFMS

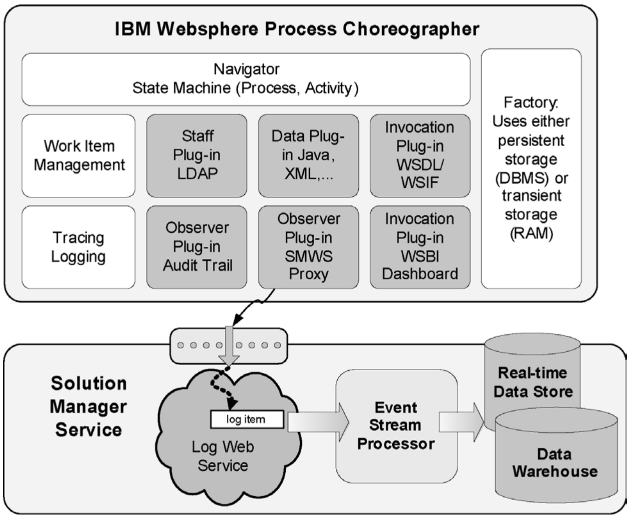
Auditing service functionality can vary significantly between WFMSs. Although the WfMC has already defined standards for workflow logs (WfMC, 1999) the major vendors of WFMSs still mostly use proprietary audit trails. Using the auditing services of WFMSs also has some advantages over the previously discussed options. If we are able to directly track BPEL4WS flows in a WFMS, we have full access to all runtime data and thereby, we can extract more contextual information about the business process.

Figure 11 shows as an example the auditing services of the IBM Websphere Process Choreographer. This WFMS logs state changes to an audit trails database table or observer plug-ins that gives direct access to audit data. An observer plug-in implements a listener interface that gets automatically invoked by the WFMS on various flow-related events, for example, when a process is started or completed, or when an activity is started or completed. It has full access to the runtime information and can decide during process execution which runtime data has to be extracted. As shown in Figure 11, an observer plug-in can directly forward the captured audit data to the *log web service* of the SMS. Although this option is the most flexible, only a few WFMSs support this type of auditing.

7.2.4 Probes in operational systems

To track business process state changes not managed by a WFMS, operational systems probes are required to sense the source system operations and generate audit data. They can be implemented at various application layers (e.g. database layer, business logic layer and presentation layer), but must be exposed to sufficient business process run-time information to perform the logging activities.

Figure 11 Observer plug-ins

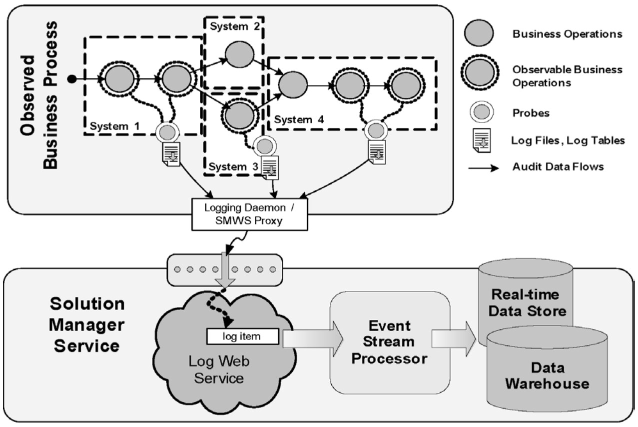


For instance, database table triggers, can automatically generate audit data each time a table record is inserted. Database triggers can, however, have a performance impact on the operational source system and sometimes do not have easy access to all runtime data of a business process (e.g. data of business objects has to be gathered from multiple tables).

If the operational source systems offer auditing mechanisms, logging daemons can extract information from the audit log files or database tables on a scheduled basis, to generate the required business process audit data. Due to the scheduled data extraction, this option causes some audit trail processing latency.

Figure 12 shows a business process supported by four operational systems with some of them that have probes. A logging daemon collects the logging data from various probes and forwards it to the SMS.

Figure 12 Operational systems with probes

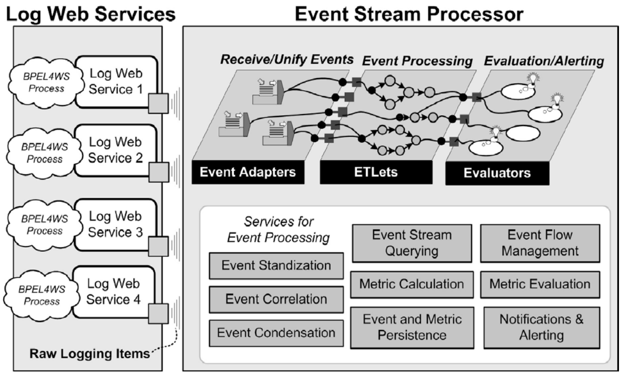


7.2.5 Event stream processor

Once the definition of the web service is an complete, logging can commence by capturing process events using one of the four methods for audit logging detailed in the previous section. After the log web service receives the log item information, it is forwarded to the ESP. The ESP has a container architecture that translates the raw logging data into standardised events and transforms these events into business performance data that is stored in the real-time data store or data warehouse. The ESP provides a robust, scalable data staging environment that supports continuous integration of a large number of events that can require complex processing logic.

Similar to Java technology for web applications, where servlets and JSPs took the place of traditional CGI scripts, our approach uses event adapters, ETLets and evaluator (see Figure 13) components that replace traditional ETL (Extraction, Transformation, Loading) solutions which very often use scripts that are hard to maintain, scale and reuse. The ESP uses *event adapters* to receive logging data and unify it into a standardised event format. *ETLets* use the standardised events as input and perform the event processing tasks, such as the computation of business metrics. ETLets also publish the calculated business metrics that can be evaluated by *evaluator* components.

Figure 13 Event stream processor



The purpose of event adapters is to receive the raw logging items from the log web service and to translate the logging items (with potentially different logging formats) into a standardised event format. Event adapters can receive raw logging items in two ways:

1. synchronously via a direct call from the log web service or
2. asynchronously via messaging software.

The second option is more scalable because it provides the ability to completely decouple the log web service from the actual processing of the logging data with the ESP. Every log web service must transmit the raw logging data to one of the event adapters from the ESP. A typical solution can have several event adapters running in parallel. They receive and dispatch events in parallel. In order to address overload situations, where not enough resources are available to instantiate ETLets for the event processing, the ESP can block an event adapter temporarily. For instance, if there is no thread available for the processing of an incoming event within a specified timeout period, the event adapter is notified of the overload situation and can react to this situation individually.

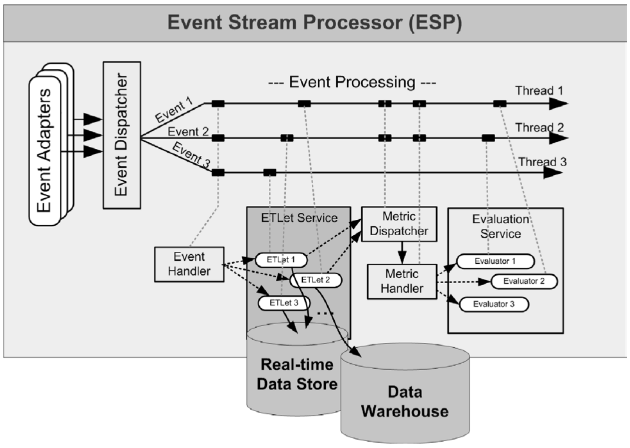
Once the event data is parsed and standardised in the event adapters, it is passed through a number of components that calculate and evaluate business metrics, and store relevant process information by loading it into dimension and fact tables of the real-time data store or the data warehouse. ETLets are similar to ETL scripts, but they run in a container of a Java application server. ETLets adhere to a set of conventions that let them run within a Java environment. With traditional ETL solutions, often a new process is started for each major type of ETL processing. Furthermore, intermediary storage is required to combine the processing tasks of an ETL process. If data extracts and data transformations are very frequent, the overhead for starting the processes and combining the processing steps can dominate the execution time. With ETLets, a lightweight Java thread, rather than a heavyweight operating system process handles each request.

The metrics calculated and published by ETLets can be evaluated by evaluator components. An evaluation of continuously calculated metrics can be very valuable because it can be used to check service levels in order to create an intelligent response (e.g. sending out notifications to business people or triggering business operations) in near real-time. Therefore, in the solution management architecture, the monitoring web service makes use of the evaluator components for listening to updated performance data. The evaluator components can be either implemented by developers or act as proxy by forwarding the evaluation requests to rule engines for more sophisticated evaluations. Evaluators are not aware of how a metric was calculated or which events triggered the metric calculation. Therefore, the ESP enables a clean separation of receiving and unifying events (event adapter), processing events (ETLets) and evaluating performance data (evaluators).

ETL scripts are not suitable for an event-driven environment where data extracts and data transformations are very small and frequent, because the overhead for starting the processes and combining the processing steps can dominate the execution time. Another limitation of ETL scripts is that they are written for a specific task in a self-sustaining manner and do not provide any kind of interfaces for data inputs and outputs. Because of this constraint of ETL scripts, we use for the ESP a container approach to manage and optimise the event processing. The ESP provides services for the execution and monitoring of event processing tasks. The core services are responsible for creating, initialising, executing and destroying the managed components that are responsible for the processing of the logging data.

The ESP handles each event with a lightweight Java thread, rather than a heavyweight operating system process. Figure 14 shows the internal event processing of the ESP. The components shown with round boxes are components that are managed by the ESP. The components shown with square boxes are internal ESP components that are used to bind all managed ESP components together. Please note that the developers never see or have to deal with the internal components. We show these internal components for illustration purposes only.

Figure 14 ESP multithreading



This approach also simplifies the programming tasks for developers who have to implement the logic for the event processing, since the ESP takes responsibility for various system-level services (such as threading, resource management, transactions, caching, persistence and so on). In our approach, we extend this concept by adding new services for the event processing, which can be leveraged by the developers. Examples of such services are the event correlation service, which can be utilised for gathering a set of related events for the processing (e.g. the calculation of the processing time requires the collection of the event pair PROCESS_STARTED and PROCESS_COMPLETED from a process instance), or the evaluation service, which significantly reduces the effort for evaluating continuously calculated process metrics.

Further details relating to the ESP component of the SMS are in McGregor and Schiefer (2003, 2004), Schiefer et al. (2003) and Schiefer and McGregor (2004).

8 Monitor web service and performance monitor

8.1 Monitor web service

The *monitor web service* is accessible during run-time and provides the proactive environment enabling users to define additional and update existing sensors to observe service level agreements and performance and monitoring objectives. These sensors constantly review service enactment performance data that is available in the data warehouse or real-time data store. Sensors can be adapted over time by the owner or partners, by reusing the *monitor web service* to update parameters.

Similarly to the functionality provided by the *define web service*, the *monitor web service* supports the general case and partner or partner group functions. The general case can only be defined by the owner of the web service.

However, in contrast to the *define web service*, both the owner and the partners can use the *monitor web service* functionality to define sensors for partners. Figure 15 shows the *monitor web service* fields when it is used in the general case.

Figure 15 Monitor web service

Name: **MonitorWebService**
Input: WebServiceDefinition
    WebServiceDescription
        Web Service ID
    Owner
        OwnerID
    Product
        ProductGroup
    PartnerGroup
        PartnerGroupID
        Name
        Description
    Measures
        ServiceLevelAgreement
            - *SLAID*
            - *SLAType*
            - *SLAValue*
            - *StartDate*
            - *EndDate*
Output:
    WebServiceDefinitionResponse
        ResultNotification (Boolean)

8.2 Performance monitor

The performance monitor component is an agent-based run-time component invoked by the *monitor web service* and enables organisations to define and change services level agreements and other selected performance measures. The performance monitor listens to performance data that is continuously generated and automatically responds to exceptional business situations, such as service level violations.

The performance monitor contains three components namely: Performance data sensors, response agents and notification agents.

The *performance data sensors* are the main controlling modules within the performance monitor. They are created, modified and deleted based on the invocation of the *monitor web service* and data driven based on the data contained within that invocation. They each individually constantly observe the data stored in data management layer that is relevant to the context for which they were created and instigate the response and notification agents as required.

The *response agents* are called by the *performance data sensors* in response to a violation in the service-level agreement and other selected performance measure monitored by that *performance data sensor* and generate an automated response to violations.

Similarly to response agents, *notification agents* are called by the *performance data sensors* when performance bounds monitored by that performance data sensor are exceeded and generate a user notification.

9 Analyse web service and analytical processor

9.1 Analyse web service

Analyse web services provides the web service interface to the Analytical Processor complement which is the reactive environment of the SMS, enabling users to analyse the web service enactment performance data that is available in the data warehouse or real-time data store. All participants in the web service may request web service instance information and access is controlled via the security policy rules set during the *define web service*. As such multiple organisations have access to the same information, one as the supplier and the rest as partners.

Analyse web services are data driven and are themselves defined to the service registry together with the *define* and *log web services* at the establishment of the SMS. The available *analyse web services* that are accessible to the owner of the web service and its partners are determined by the information supplied when the *define web service* is used to define their web service to the Solution Management Service. The more information supplied to the *define web service*, the more *analyse web services* made available.

The performance data accessible through the Analytical Processor via the *analyse web services* relates to web service enactments and provides information such as process metrics (e.g. processing times, costs and rework) and service level-violations.

Suppliers and partners are then able to use the analytical processing interface made available by the Solution Management Service to display the information in a user-friendly format. Alternatively they would be able to use the existing On-Line Analytical Processing (OLAP) tools, within their organisation, to format and display the XML received.

9.2 Analytical processor

The agent-based analytical processor operates during run-time when web services instances are enacted and information about the enactments has been supplied to the ESP via the *log web service*.

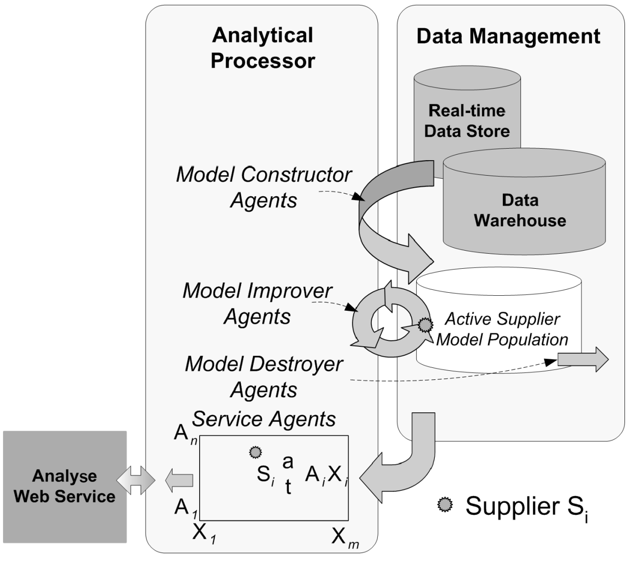
All *analyse web services* are serviced by service agents that extract data from the data management layer. In addition, the analytical processor provides agents that work on the population of organisational models that are stored within the data management layer. These models are created, improved or destroyed, over time by the analytical processor agents.

These agents are designed based on the Asynchronous Team (A-Team) problem solving architecture (Rachlin et al., 1999). The agents work on models stored in the data management layer. The model constructor agents create new models to initialise the population. Model improver agents improve the realism these models as more and more data becomes available in the data warehouse, as more web services instances are enacted. Models that are deemed inactive are removed or disabled from the model population. Thus, the model population serves as a repository for the organisation's growing knowledge of its own behaviour and the behaviour of its partners (McGregor and Kumaran, 2002).

Figure 16 illustrates the analytical processor architecture within the context of a supplier behaviour model. In this example, multiple suppliers perform the supplier role within the web service that has been defined to the SMS. A model for supplier behaviour has been established with two behaviour attributes (A and X). Models can be established with more than two attributes and models can be created/changed/deleted over time to cater for flexible business needs. Data constructor agents extract initial information about suppliers from the data warehouse and use that information to determine attribute values for each supplier and hence the position of each supplier within the attribute space. At periodic intervals, model improver agents extract up to date information on the suppliers from the Data Warehouse and use that information to reposition each supplier within the attribute space. As a result, suppliers exhibiting similar attribute behaviours can be grouped to form behavioural clusters. Suppliers that are deemed to be inactive are periodically removed from the model population by model destroyer agents. Service agents provide information relating to the supplier model in response to requests received via *analyse web services*.

Further details relating to the Analytical Processor component of the SMS are in McGregor and Kumaran, (2002a,b) and McGregor (2002).

Figure 16 Supplier behaviour model



10 Example – monitoring of a supply chain process

In this section, we present a supply chain management scenario, which we extend with SMS-specific components. The provision of goods and services to consumers and the reverse flow of used goods create a network of organisations that needs to be coordinated in order to function efficiently. While enterprise management functions are focused on a single organisation, the management of supply chains extends this view to cover neighbouring enterprises, up to the supply chain in its entirety. The SMS shown in Figure 17 efficiently supports this widened scope and provides supply chain managers with performance data about their own organisation and business partners.

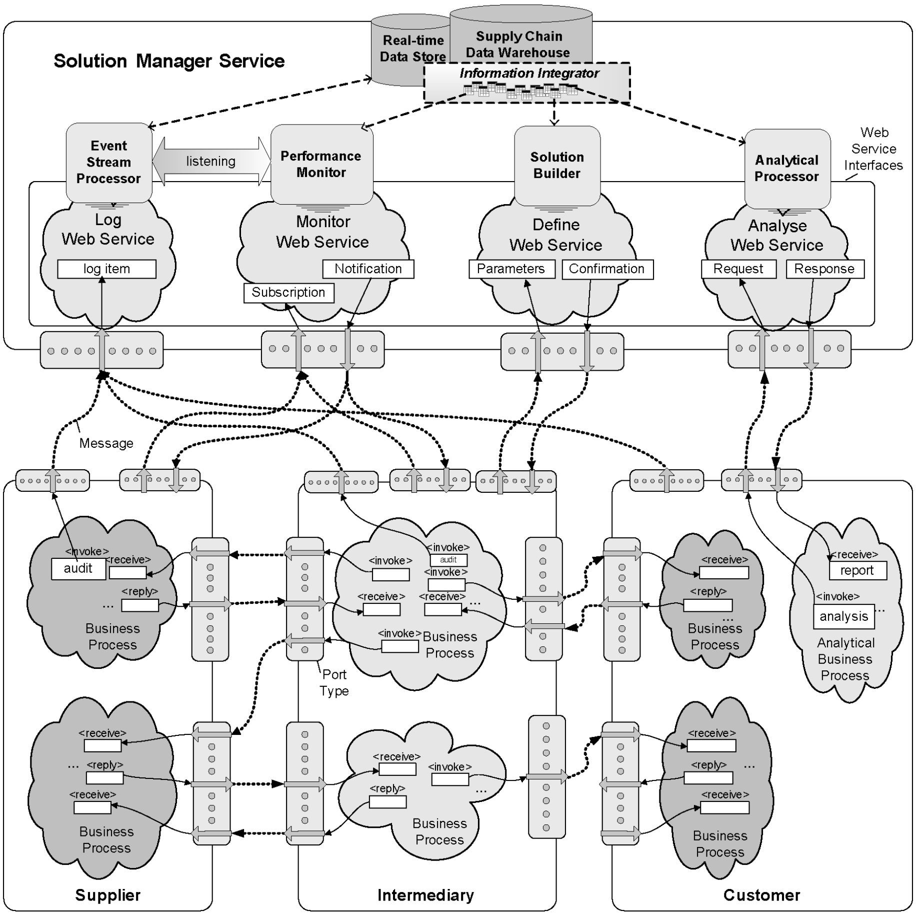
In our example, the intermediary organisation could be a wholesaler or retailer that serves as the owner of the monitoring infrastructure at the same time. The intermediary organisation uses the *define web service* to capture the web service states and state transition rules of the supply chain, to define the performance objects such as warehouse costs, costs per order or order processing time (among others) and to define access policies for clients that are allowed to monitor and analyse the performance data of the supply chain.

For this example, we assume that the supplier and intermediary organisation are monitoring clients and want to be notified by triggering a certain web service when a performance indicator is out of bounds. In order to define the service levels for these performance indicators, the monitoring clients subscribe to the *monitor web service* and define service level information, which includes the thresholds for the monitored metrics and also the web service operations that should be invoked when a service level is violated. All organisations have access to the *analysis web service*, to retrieve recent and historic performance data based on the security policies defined by the intermediary organisation.

The participating organisations have to extend their business processes with additional auditing steps which are automatically invoked by the workflow engine or the operational system when transactions are performed within the supply chain. Each time an organisation makes a *log web service* request to the solution manager, the ESP receives the logging data and transforms it into event and performance data and stores them in the real-time data store and data warehouse. The ESP also manages evaluator components which listen to the continuously calculated metrics. The performance monitor utilises this mechanism to check the service levels each time a new metric is computed. In the case of a performance indicator violating a service level, the monitor web service will trigger a web service operation of the monitoring clients.

Though not reflected in this example, the supplier can utilise the *analyse web service* and the customer can utilise the *monitor web service*.

Figure 17 Supply chain monitoring with the solution manager



11 Conclusion and future work

This paper has presented an infrastructure for a shareable, web service based IDSS for on-demand business process management, called the SMS that collects information about business processes in a centralised repository, to share among authorised parties. The interaction with this IDSS, is via a set of web services and the infrastructure boasts several innovative agent-based modules. Data-driven security policies maintain appropriate data access. A supply chain example was used to demonstrate the benefits of this approach.

The following limitations of the presented approach are considered for future research:

The current architecture assumes the existence of only one solution manager per process. We believe that a tailored and layered solution manager architecture can be defined where context-specific solution managers with their own data marts exist supporting different functional needs.

Further development is required to enable automated feedback of recommendations to organisational systems based on recommended changes to process definition structure, etc. The effect of these changes on actual business process performance needs to be evaluated further.

Finally, while robust prototypes for the log web service and ESP have been developed, the prototypes for the remaining components are at earlier levels of maturity and hence the development of a commercial grade version will be pursued in future work.

References

Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I. and Weerawarana, S. (2003) ‘Business process execution language for web services, version 1.1’, *OASIS 2003-05-05*.

Bonifati, A., Casati, F., Dayal, U. and Shan, M-C. (2001) ‘Warehousing workflow data: challenges and opportunities’, *Presented at VLDB01*, Rome, Italy.

Cachon, G.P. and Fisher, M. (1999) ‘Supply chain inventory management and the value of shared information’, *Management Science*, Vol. 46, pp.1032–1048.

Casati, F., Castellanos, M., Dayal, U., Hao, M., Shan, M-C. and Sayal, M. (2002) ‘Business operation intelligence research at HP labs’, *Data Engineering Bulletin*, Vol. 25, pp.32–35.

Deming, W.E (1992) *Out of the Crisis: Quality, Productivity and Competitive Position*, Cambridge: Cambridge University Press.

Derzteler, G. (2000) *Prozeßmanagement auf Basis von Workflow-Systemen: Ein integrierter Ansatz zur Modellierung, Steuerung und Überwachung von Geschäftsprozessen*, Lohmar, Köln: Josef Eul Verlag.

Dunn, R.H. (1990) *Software Quality: Concepts and Plans*, Englewood Cliffs, NJ: Prentice Hall.

Foster, I. (2003) ‘The grid: computing without bounds’, *Scientific American*, pp.61–67.

Gavirneni, S., Kapuscinski, R. and Tayur, S. (1998) 'Value of information in capacitated supply chains', *Management Science*, Vol. 45, pp.16–24.

Hammer, M. (2001a) 'Processed change', *Journal of Business Strategy*, Vol. 1, pp.11–15.

Hammer, M. (2001b) 'The superefficient company', *Harvard Business Review*, Vol. 79, pp.82–91.

Harel, D. (1988) 'On visual formalisms', *Communications of the ACM*, Vol. 31, pp.514–530.

Hollingsworth, D. (1995) 'The workflow reference model version 1.1', *Workflow Management Coalition*, Winchester, UK WFMC-TC-1003, 19 January.

Holten, R., Dreiling, A., zur Muehlen, M. and Becker, J. (2002) 'Enabling technologies for supply chain process management', *Presented at 2002 Information Resource Management Association International Conference*, Seattle, Washington.

IDS Scheer AG (2000) 'Process performance manager', *Saarbrücken*, White Paper 2002-01-17.

Jeng J.J. and Schiefer, J. (2003) 'An agent-based architecture for analyzing business processes of real-time enterprises', *Presented at Proceedings of the Seventh International Enterprise Distributed Object Computing Conference (EDOC 2003)*, Brisbane, Australia.

Kazman, R., Bass, L., Abowd, G. and Webb, M. (1994) 'SAAM: a method for analyzing the properties of software archietectures', *Presented at International Conference on Software Engineering – ICSE'16*, Sorrento, Italy.

Kontzer, T. (2003) 'Outsmarting spam', *Information Week*.

Kueng, P. (1998) 'Supporting BPR through a process performance measurement system', in P. Banerjee et al. (Eds). *Business Information Technology Management*, New Delhi: Har-Anand Publications, pp.422–434.

Kueng, P. (2000) 'Process performance measurement system: a tool to support process-based organizations', *Total Quality Management*, Vol. 11, pp.67–85.

Kueng, P. and Krahn, A. (1999) 'Building a process performance measurement system – some early experiences', *Journal of Scientific and Industrial Research*, Vol. 58, pp.149–159.

Lambros, P., Schmidt, M. and Zentner, C. (2001) 'Combine business process management technology and business services to implement complex web services', http://www-4.ibm.com/software/ solutions/ webservices/pdf/ BPM.pdf, accessed 4th August, 2006.

Lee, H.L., So, K.C. and Tang, C.S. (1999) 'The value of information sharing in a two-level supply chain', *Management Science*, Vol. 46, pp.626–643.

Leymann, F. and Altenhuber, W. (1994) 'Managing business processes as an information resource', *IBM Systems Journal*, Vol. 33, pp.326–348.

Leymann, F. and Roller, D. (2000) *Production Workflow: Concepts and Techniques*, Upper Saddle River, NJ: Prentice Hall.

McGregor, C. (2002) 'IW-MONS: a methodology for the design of intelligent workflow monitoring systems', *Faculty of Information Technology. Sydney: University of Technology*, Sydney.

McGregor, C. (2002) 'The impact of business performance monitoring on wfmc standards', L. Fischer (Ed). *Workflow Handbook 2002, Lighthouse Point (FL): Future Strategies*.

McGregor, C. (2003) 'A method to extend BPEL4WS to enable business performance measurement', *Presented at 1st International Conference on Web Services ICWS'03*, Las Vegas.

McGregor, C. and Edwards, J. (2000) 'Intelligent workflow monitoring systems', *Presented at Proceedings of the Doctoral Consortium at the 11th Australasian Conference on Information Systems (ACIS 2000)*, Brisbane.

McGregor, C. and Kumaran, S. (2002a) 'An agent-based system for trading partner management in B2B e-commerce', *Presented at Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems, 2002. RIDE-2EC 2002. Proceedings of Twelfth International Workshop on 2002*.

McGregor, C. and Kumaran, S. (2002b) 'Business process monitoring using web services in B2B e-commerce', *Presented at Parallel and Distributed Processing Symposium, Proceedings International of IPDPS 2002*, Abstracts and CD-ROM, 2002.

McGregor, C. and Schiefer, J. (2003) 'A framework for analyzing and measuring business performance with web services', *Presented at E-Commerce, 2003. CEC 2003. IEEE International Conference on 2003*.

McGregor, C. and Schiefer, J. (2004) 'A web-service based framework for analyzing and measuring business performance', *Information Systems and e-Business Management*, Vol. 2, pp.89–110.

McLellan, M. (1996) 'Workflow Metrics – one of the great benefits of workflow management', in H. Oesterle and P. Vogler (Eds). *Praxis des Workflow-Management*, Braunschweig: Vieweg, pp.301–318.

Rachlin, J., Goodwin, R., Murthy, S., Akkiraju, R., Wu, F., Kumaran, S. and Das, R. (1999) 'A-teams: an agent architecture for optimization and decision support', *Intelligent Agents V: Agent Theories, Architectures and Languages*, Vol. 1555, pp.261–276.

Raufer, H. (1997) *Dokumentenorientierte Modellierung und Controlling von Geschäftsprozessen. Integriertes Workflow-Management in der Industrie*, Wiesbaden: Deutscher Universitäts-Verlag.

Salimifard, K. and Wright, M. (2001) 'Petri net-based modelling of workflow systems: an overview', *European Journal of Operational Research*, Vol. 134, pp.664–676.

Sarbanes-Oxley Act of 2002 (2002) *Pub L No. 204, 116 Stat 745 (Codified at 15 USC 7201 et seq.), Congress of the United States of America*, One Hundred Seventh ed.

Sayal, M., Casati, F., Dayal, U. and Shan, M-C. (2002) 'Business process cockpit', *Presented at VLDB 2002*, Hong Kong, China.

Schiefer, J. and McGregor, C. (2004) 'Correlating events for monitoring business processes', *Presented at Sixth International Conference on Enterprise Information Systems*, Porto.

Schiefer, J., Jeng, J-J. and Bruckner, R.M. (2003) 'Real-time workflow audit data integration into data warehouse systems', *Presented at 11th European Conference on Information Systems*, Naples.

van der Aalst, W.M.P. and van Hee, K. (2002) *Workflow Management. Models, Methods, and Systems*, Cambridge, MA: MIT Press.

van der Aalst, W.M.P. and van Dongen, B.F. (2002) 'Discovering workflow performance models from timed logs', *Presented at International Conference on Engineering and Deployment of Cooperative Information Systems (EDCIS 2002)*, Berlin.

Weiß, D. (1998) *Prozeßkostenrechnung und Workflow Management: Konzeption und Umsetzung eines Schnittstellensystems*, Wiesbaden: Deutscher Universitäts-Verlag.

Weiß, D. (1999) 'Wie kann das Workflow Management die Prozeßkostenrechnung unterstützen?' *Controlling*, Vol. 11, pp.543–549.

Weiß, D. and Zerbe, S. (1995) 'Prozeßkostenrechnung und Vorgangssteuerung' *Controlling*, Vol. 7, p.1.

WfMC (1998) 'Workflow audit trail (interface 5) application programming interface specification v 0.1', *Workflow Management Coalition*, Winchester, UK, Document Number WFMC-TC-1022, 2 October.

WfMC (1999) 'Audit data specification, version 2', *Workflow Management Coalition*, Winchester, ID, Document Number WFMC-TC-1015.

WfMC (2002) 'Workflow process definition interface – XML process definition language: document status – 1.0 final draft', *Workflow Management Coalition*, Lighthouse Point, FL, Document Number WFMC-TC-1025 2002-10-25.

zur Muehlen, M. (2000) 'Workflow-based process controlling – or: what you can measure you can control', in L. Fischer (Ed). *Workflow Handbook 2001,* Lighthouse Point, FL: Future Strategies, pp.61–77.

zur Muehlen, M. (2001) 'Process-driven management information systems – combining data warehouses and workflow technology', *Presented at Proceedings of the 4th International Conference on Electronic Commerce Research (ICECR-4)*, Dallas, TX.

zur Muehlen, M. (2003) *Workflow-based Process Controlling. Foundation, Design, and Implementation of Workflow-driven Process Information Systems*, Berlin: Logos, Vol. 6.

zur Muehlen, M. and Rosemann, M. (2000) 'Workflow-based process monitoring and controlling – technical and organizational issues', *Presented at Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS 2000)*, Wailea, HI.