Grant Agreement N° 215483

Title: *Survey on Business Process Management*

Author: *V. Andrikopoulos (Tilburg), S. Benbernou (UCBL), M. Bitsaki (UOC), O. Danylevych (USTUTT), M.S. Hacid (UCBL), W.J. van den Heuvel (Tilburg), D. Karastoyanova (USTUTT), B. Kratz (Tilburg), F. Leymann (USTUTT), M. Mancioppi (Tilburg), K. Mokhtari (UCBL), C.N. Nikolaou (UOC), M.P. Papazoglou (Tilburg), B. Wetzstein (USTUTT)*

Editor: *W.J. van den Heuvel (Tilburg)*

Reviewers: *Salima Berbernou (UCBL)*

Mariagrazia Fugini (POLIMI)

Identifier: *Deliverable # PO-JRA-2.1.1*

Type: *Deliverable*

Version: *1*

Date: *14-07-2008*

Status: *Final*

Class: *External*

Management Summary

In its simplest form, Business Process Management (BPM) is a suite of software technologies focusing on the management of the complete lifecycle of a business process. To date, most BPM deployments have been narrow in scope, adopting an organization-centric view and providing only improvements in specific business functions. As a result, current BPM suites only enable organizations to enhance their existing processes. The next-generation of service-enabled BPM will serve as a means of developing mission-critical applications based on strategic technology capable of creating and executing cross-enterprise collaborative business processes and business-aware transactions, so that organizations can deploy, monitor, and continuously update cross-enterprise functions within a mixed environment of people, content, and systems. Such collaborative, complex end-to-end service interactions give raise to the concept of Agile Service Networks. In this report, we assess the state-of-the-art in BPM, surveying the basic concepts, describe the features, techniques and enabling technologies necessary for making BPM a reality and explain the need for service-based BPM. The report also highlights the need for moving from a relatively static and organization-centric view of BPM to a much more dynamic, high-value one based on Agile Service Networks.

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement n° 215483 (S-Cube).

Members of the S-CUBE consortium:

University of Duisburg-Essen	Germany
Tilburg University	Netherlands
City University London	U.K.
Consiglio Nazionale delle Ricerche	Italy
Center for Scientific and Technological Research	Italy
The French National Institute for Research in Computer Science and Control	France
Lero - The Irish Software Engineering Research Centre	Ireland
Politecnico di Milano	Italy
MTA SZTAKI – Computer and Automation Research Institute	Hungary
Vienna University of Technology	Austria
Université Claude Bernard Lyon	France
University of Crete	Greece
Universidad Politécnica de Madrid	Spain
University of Stuttgart	Germany

Published S-CUBE documents

These documents are all available from the project website located at http://www.s-cube-network.eu/

The S-CUBE Deliverable Series

Vision and Objectives of S-CUBE

The Software Services and Systems Network (S-Cube) will establish a unified, multidisciplinary, vibrant research community which will enable Europe to lead the software-services revolution, helping shape the software-service based Internet which is the backbone of our future interactive society.

By integrating diverse research communities, S-Cube intends to achieve world-wide scientific excellence in a field that is critical for European competitiveness. S-Cube will accomplish its aims by meeting the following objectives:

- Re-aligning, re-shaping and integrating research agendas of key European players from diverse research areas and by synthesizing and integrating diversified knowledge, thereby establishing a long-lasting foundation for steering research and for achieving innovation at the highest level.
- Inaugurating a Europe-wide common program of education and training for researchers and industry thereby creating a common culture that will have a profound impact on the future of the field.
- Establishing a pro-active mobility plan to enable cross-fertilisation and thereby fostering the integration of research communities and the establishment of a common software services research culture.
- Establishing trust relationships with industry via European Technology Platforms (specifically NESSI) to achieve a catalytic effect in shaping European research, strengthening industrial competitiveness and addressing main societal challenges.
- Defining a broader research vision and perspective that will shape the software-service based Internet of the future and will accelerate economic growth and improve the living conditions of European citizens.

S-Cube will produce an integrated research community of international reputation and acclaim that will help define the future shape of the field of software services which is of critical for European competitiveness. S-Cube will provide service engineering methodologies which facilitate the development, deployment and adjustment of sophisticated hybrid service-based systems that cannot be addressed with today's limited software engineering approaches. S-Cube will further introduce an advanced training program for researchers and practitioners. Finally, S-Cube intends to bring strategic added value to European industry by using industry best-practice models and by implementing research results into pilot business cases and prototype systems.

S-CUBE materials are available from URL: http://www.s-cube-network.eu/

Table of Contents

Table of illustrations

List of acronyms

ACID	Atomicity, Consistency, Isolation, Durability
ASAP	Accelerated SAP
ASN	Agile Service Network
B2B	Business-to-Business
BPI	Business Process Integration
BPM	Business Process Management
BPMN	Business Process Modelling Notation
BPO	Business Process Ontology
CEP	Complex Event Processing
CFO	Chief Financial Officer
EAI	Enterprise Application Integration
ebXML	Electronic Business using eXtensible Markup Language
EPC	Event Process Chains
GUI	Graphical User Interface
KPI	Key Performance Indicator
OGSA	Open Grid Services Architecture
OWL	Web Ontology Language
QoS	Quality of Service
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
TPBP	Timed web service Business Protocol
UML	Unified Modelling Language
WS-AT	Web Services Atomic Transaction
WS-BA	Web Services Business Activity
WS-BPEL	Business Process Execution Language
WSCI	Web Service Choreography Interface
WSCL	Web Service Conversation Language
WSDL	Web Service Description Language
WS-Transaction	Web Service Transaction
XCBL	XML Common Business Library

1 Overview

In order to gain competitive advantage, enterprises are striving to become electronically connected to their customers, suppliers, and partners. Therefore, they are integrating a wide range of discrete business processes across application boundaries of all kinds. Application boundaries may range from simple enquiries about a customer's order involving two applications, to complex, long-lived transactions for processing an insurance claim involving many applications and human interactions. When integrating on such a scale, enterprises need a greater latitude of functionality to overcome multiple challenges arising from the existence of proprietary interfaces, diverse standards, and approaches targeting automation of transactions and business processes, process analysis and visualization. In addition, and perhaps even more challenging, integration of processes needs to be modifiable to be able to continuously optimize their performance. Such challenges are addressed by Business Process Management (BPM).

Business Process Management has recently emerged as both a management principle and a suite of software technologies focusing on management of the lifecycle of a business process ranging from business goals reflected in the definition of business processes, to the deployment, execution, measurement, analysis, change, and redeployment of these business processes. To date, most BPM deployments have been narrow in scope, adopting an organization-centric view providing only improvements in specific business functions, and as a result current BPM suites only enable organizations to enhance their existing processes. This is extremely restrictive for applications characterized by wide-scale and complex dynamic interactions.

The next-generation of service-enabled BPM will serve as a means of developing mission-critical applications based on strategic technology capable of creating and executing cross-enterprise collaborative business processes and business-aware transactions, and connecting entire business value chains. The trend will be to move from a relatively static view of an organization to a much more dynamic, high-value one where business process interactions and trends are examined much more closely to understand much more accurately the business dynamics. The next-generation of service-enabled BPM will rely upon powerful analysis capabilities to provide an effective approach for targeting business problems in areas like compliance, change management, quality improvement, and operational business health, delivering more business value and reducing risk. Such collaborative, complex end-to-end service interactions give raise to the concept of Agile Service Networks (ASNs).

In this report, we assess the state-of-the-art in BPM and survey its basic concepts, describe the features, techniques and enabling technologies, and explain the need for service-based BPM. The report emphasizes the interplay between BPM and Service Oriented Architectures (SOAs) focusing on how to leverage SOA for achieving effective BPM. It also highlights the need for moving from a relatively static and organization-centric view of BPM to a much more dynamic, high-value one. This BPM view is based on strategic Agile Service Network technology capable of creating and executing cross-enterprise collaborative business processes and business-aware transactions so that organizations can deploy, monitor, and continuously update cross-enterprise functions within a mixed environment of people, content and systems.

The report is organized as follows: section 2 focuses on the notion of business processes; it discusses their major characteristics, presents how they can be expressed via means of workflows, and examines process reusability through patterns. Section 3 subsequently introduces the concepts of business protocols and business-aware transactions, reviews contemporary approaches, and ponders open research issues with regard to their realization. In particular, this section concentrates on classifying, specifying, evolving, and reasoning about business protocols. It also presents the notion and main characteristics of business-aware transactions and explains how they relate to current research activities in the Web services field. Section 4 provides a broad perspective on BPM, discussing the different phases of the BPM lifecycle, and in particular focusing on business process modeling, business process execution, and business process monitoring and analysis. Furthermore, it examines the roles and responsibilities of various stakeholders with respect to the BPM lifecycle, and the beneficial interplay of SOA and BPM. Section 5 introduces the notion of Agile Service Networks and describes their role with respect to BPM. It also explores the relationship between Key Performance Indicators found within ASNs, and contractual business process requirements. This relationship is used for the monitoring of SOA-enabled business processes. The design and development of ASNs is also examined, in particular with regard to the evolutionary aspect of these networks. Additionally, the usefulness of simulation techniques and tools in analysing and adapting ASNs is highlighted. Finally, section 6 summarizes this report.

2 Business Processes

A *process* is an ordering of activities with a beginning and an end; it has inputs (in terms of resources, materials and information) and a specified output (the results it produces) [91]. We may thus define a process as any ordered set of steps that is initiated by an event, transforms information, materials, or commitments, and produces some output [92]. A *business process* is a process used to achieve a well-defined business outcome and is completed according to a set of procedures. The key elements in this definition are that a business process may span organizations and may typically involve both people and systems.

A (business) process view implies a horizontal view on a business organization and looks at processes as sets of interdependent activities designed and structured to produce a specific output for a customer or a market [92]. A business process defines the results to be achieved, the context of the activities, relationships between the activities, and the interactions with other processes and resources, and users. A business process may receive events that alter the state of the process and the ordering of activities. A business process may produce events for input to other applications or processes. It often invokes applications to perform computational functions, and it may post assignments to human work lists to request actions by human actors. Business processes can be measured, and different performance measures apply, like cost, quality, time and customer satisfaction.

Each enterprise has unique characteristics and procedures that are embedded in its business processes. Most enterprises perform a similar set of repeatable routine activities that may include the development of manufacturing products and services, bringing these products and services to market and satisfying the customers who purchase them. *Automated business processes* can perform such activities. We may view an automated business process as a sequence of activities precisely *choreographed* (i.e., coordinated according to a model of the overall structure of the process), and systematically directed towards performing a certain business task and bringing it to completion herby producing a desired business outcome.

A business process can be simple, e.g. order fulfillment, or complex, e.g. new product development, short-running, e.g., calculating taxes or revenues, or long-running, e.g. regulatory compliance, function-specific, e.g., proposal management, or industry-specific (e.g. energy procurement). It can exist within a single department, e.g. transportation, run throughout the entire enterprise, e.g. strategic sourcing, or extend across the whole value chain, e.g. supply chain management. Examples of typical processes in manufacturing firms include amongst other things new product development (which cuts across research and development, marketing and manufacturing), customer order fulfillment (which combines sales, manufacturing, warehousing, transportation, and billing) and financial asset management. The possibility to design, structure, measure processes and determine their contribution to customer value, makes them an important starting point for business improvement and innovation initiatives.

The largest possible process in an organization is the *value chain*. The value chain is decomposed into a set of core business processes and support processes necessary to produce a service, product or product line. These core business processes are subdivided into activities. An *activity* is an element that performs a specific function within a process. Activities can be as simple as sending or receiving a message, or as complex as coordinating the execution of other processes and activities. A business process may encompass complex activities some of which run on back end systems such as, for example, a credit check, automated billing, a purchase order, stock updates and shipping, or even such activities as sending a document, and filling a form. An activity may invoke another business process in the same or a different business system domain. Activities will inevitably vary greatly from one company to another and from one business analysis effort to another.

At runtime, a business process definition may have *multiple instantiations*; each operating independently of the other and each instantiation may have multiple activities that are concurrently active. A process *instance* is a thread of activities that is being enacted (managed) by a workflow engine (a.k.a. process engine) based on a business process definition (a.k.a. business process model). In general, instances of a process, its current state and the history of its actions will be visible at run time and expressed in terms of the business process definition so that users can determine the status of activities and business specialists can monitor the activity and identify potential improvements to the business process definition.

2.1 Characteristics of Business Processes

A business process is typically associated with operational objectives and business relationships, for example an insurance claims process, or a collaborative engineering development process. A process may be wholly contained within a single organizational unit or may span several different organizations, such as in a customer-supplier relationship. Typical examples of processes that cross organizational boundaries are purchasing and sales processes jointly set up by buying and selling organizations. The Internet is now a trigger for the design of new business processes and the redesign of existing ones. New expectations have come up with setting up Web services, which aim at the design of standardized business process based solutions.

Every process has a customer. The customer may be external, like the final customer for whom a service or product is produced, or internal, like another process for which the output of the process under consideration forms an input. Not every process is directly triggered by a customer order. It is possible that a process is triggered by a standard procedure (event). For example, salary payments are triggered by a specific date in the month.

Every business process implies processing: a series of activities (processing steps) leading to some form of transformation of data or products for which the process exists. Transformations may be executed manually or in an automated way. A transformation will encompass multiple processing steps. For example the process ‘authorizing invoices’ will encompass the steps ‘checking whether the invoice has not been paid yet’, ‘checking the agreed purchasing conditions’, ‘checking the receiving report’, ‘checking calculations’ and ‘checking name, address, and bank account of the creditor’. If and only if all the checkpoints are correct, the invoice will be registered in the accounts payable administration.

Processes have *decision points*. Decisions have to be made with regard to routing and allocation of processing capacity. In a highly predictable and standardized environment, the trajectory in the process of a customer order will be established in advance in a standard way. Only if the process is complex and if the conditions of the process are not predictable, routing decisions have to be made on the spot. In general the customer orders will be split into a category that is highly proceduralized (and thus automated), and a category that is complex and uncertain. In the latter case, human experts will be needed and manual processing is a key element of the process.

Finally, every process delivers a product, like a mortgage or an authorized invoice. The extent to which the end product of a process can be specified in advance and can be standardized impacts the way that processes can be structured and automated. In many situations the simple equation “product =

process" holds, i.e. the process that is executed to produce a product is the point of emphasis of the producing company, and the product produced by means of the process is the point emphasis of the customer. This equation is key to understanding why process technology is so important: the speed, cost, quality etc. of the process correlates to the speed, cost, quality etc. of the product produced by means of the process. Similarly, the need for flexibility and adaptabilty of processes is also evident based on this equation: competitive pressure requires new or modified products to be rolled out fast and efficiently, which means that new processes must be defined or existing processes must be changed fast, and efficiently.

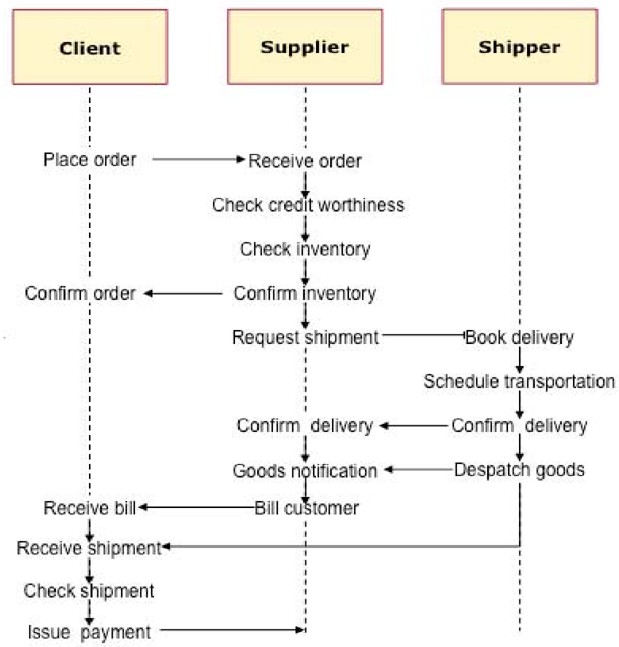

Figure 1 Order management process flow diagram [102]

A simplified version of an order management business process is depicted in Figure 1. Activities in the order management process include receiving the sales order, allocating inventory, shipping products, billing, and making sure that the payment is received. Some of these processes may execute for long periods of time, while others may execute in milliseconds. Several activities are executed at the supplier's site such as checking the credit worthiness of the customer, determining whether or not an ordered part is available in the product inventory, calculating the final price for the order and billing the customer, selecting a shipper, and scheduling the production and shipment for the order.

All steps in this figure involve a process-to-process coordination and conversation, with customized alerts set up across the network to track exceptions and provide manual intervention if necessary.

In the previous example, the business order management process describes how a product is sent from a supplier to a client. An instance of this process is actually dispatching a product(s) to a specific client. The instance of a process comprises of activity instances that include the actual work items that are passed to a workflow participant – the client, supplier, shipper roles - within this activity for action, or to another process for action. For instance, in the order management process a specific shipping company may receive all shipment related documents for a specific product and may be asked to come up with specific shipment date and shipping price for the proposed shipment.

To summarize this discussion, a business process may be characterized by the following behavior:

- It may contain defined conditions triggering its initiation in each new instance (e.g. the arrival of a claim) and defined outputs at its completion.
- It may involve formal or relatively informal interactions between participants.
- It may contain a series of automated activities and/or manual activities, which lie outside the scope of workflow management. Activities may be large and complex, involving the flow of materials, information and business commitments.
- It exhibits a very dynamic nature so they can respond to demands from customers and to changing market conditions.
- It is widely distributed and customized across boundaries within and between organizations, often spanning multiple applications with very different technology platforms.
- It has a duration that may vary widely. A core business process is usually long running – a single instance of a process such as order to cash may run for months or even years.

2.2 Business Rules and Policies

Business rules are important in the modeling of business processes, in that they help define the business terms and facts (structural assertions) as well as the constraints underlying the business behavior (action assertions). Business rules represent core business policies that capture the nature of an enterprise's business model and define the conditions that must be met in order to move to the next stage of the process. *Business rules* are represented as compact (declarative) statements about an aspect of the business that can be expressed within an application in unambiguous terms that can be directly related to the business and its collaborators and as such they determine the route of action to be followed [94], [95].

Business rules can be viewed as business directives in support of business policies. Business rules enable business analysts to define, update, and manage key decisions and policies governing business processes and applications, e.g. business policies within business processes that are likely to change can be captured using business rules. For instance, for a healthcare application, business rules may include policies on how new claim validation, referral requirements, or special procedure approvals are implemented. Business rules can represent among other things typical business situations such as escalation ("send this document to a supervisor for approval"), managing exceptions ("this loan is more than $500K, send it to the CFO"), or progression assurance ("make sure that we deal with this within 30 minutes or as specified in the customer's service-level agreement").

Business rules can be integrated with enterprise applications so that they can be used for business decision making, using ordinary business data. Business rules, in general, automate and facilitate business processes. They allow business analysts and even users to create, understand and maintain the rules and policies of the business and associate them with relevant business processes. They are usually grouped into independent but chainable rule-sets and perform inferences within and over such rule-sets.

In the traditional application structure business rules are buried in the application while in a more modern approach they are separated. Just as process flow can be separated from application code into an external BPM engine, the same can be done with business rules. Separating both process definitions and business rules empowers a business analyst to make operational changes more quickly, providing maximum flexibility and adaptability. Defining business rules as separate artifacts enables their reuse even across process definition. This is key since business rules may become complex and represent decisions that have to be consistent across processes. By referencing business rules in decision points from within process definition this can be ensured.

Over the last few years several process-based composition languages have emerged, such as the Business Process Execution Language or BPEL (see section 4.3.2 and PO-JRA-2.2.1) and the Business Process Modeling Language (BPML). These languages define the composition on the basis of a process that specifies the control and data flow among the services to be composed. To achieve business process support that is flexible and compliant to business policy and regulations, an architectural context is required in which business rules are enforced during business process enactment. In this approach, the whole business logic underlying the composition including business policies and constraints is coded as a monolithic block. As a result, business rules are hard to change without affecting the core composition logic.

Charfi and Mezini [115] propose a hybrid composition approach where the composition logic is broken down into a core part (the process) and several well-modularized business rules that exist and evolve independently. They also discuss two alternative technologies for implementing business rules in encapsulated units, using aspects and a rule-based engine. This approach allows for a more modular and flexible web service composition.

Goedertier and Vanthienen [119] present an analysis regarding the role of business rule modeling in achieving business process flexibility. In particular, it is argued that flexible business process models require business rules as a declarative formalism to capture the semantics of policy and regulation. Different kinds of business rules can be used to generate less complex control-flow-based business process models.

Keeping business processes and business rules separate at modeling time, raises the problem of how to link the enforcement of business rules, the manipulation of data, and the enactment of processes at execution time. SOA techniques can be used to integrate the functionality of different applications, offering process support and business rule support while maintaining a strong de-coupling [116]. Rosenberg and Dustdar [117] show how business rules can be integrated in BPEL using a rule interceptor service that intercepts each incoming and outgoing BPEL web service call to automatically apply business rules. Business rules are also present in the Business Collaboration Development Framework (BCDF) [118]. This framework strives for adaptability in business collaboration through web services using development rules – which include business rules – for domain analysis, management rules for validation and verification, and derivation rules for model transformation.

As organizations strive to meet compliance agendas, there is an evident need to provide systematic approaches that assist in the understanding of the interplay between (often conflicting) business and control objectives during business process design. There have been recently some efforts towards support for business process modeling against compliance requirements.

In particular the work of zur Muehlen and Rosemann [124] provides an appealing method for integrating risks in business processes. The proposed technique for “risk-aware” business process models is developed for EPCs (Event Process Chains) using an extended notation. Similarly [123] present a logical language, PENELOPE, that provides the ability to verify temporal constraints arising from compliance requirements on effected business processes. Significant research exists on the modeling of control flow in business processes, particularly in the use of patterns to identify commonly used constructs. For instance, [122] provide temporal rule patterns for regulatory policies, although the objective of this work is to facilitate event monitoring rather than the usage of the patterns for support of design time activities. Finally, [120] presents a research agenda in the space of business process compliance, identifying major technical and organizational challenges and then address the effective modeling of control objectives and their eventual propagation onto business process models.

Other than the research activities described in the context of business process modeling against compliance requirements, the challenges of compliant business processes have so far received only limited attention and the issue has never been addressed to its entirety. For a holistic approach to compliant business process management (one that covers the entire compliance life-cycle from design time checking to run-time monitoring and adaptation of services), many research themes have to be addressed [126], including: (1) advanced mechanisms for continuously auditing service-based applications, (2) understanding the need for changes of business processes due to compliance issues and its potential deep effects [125], (3) validating compliance at runtime (e.g., by means of monitoring) and providing remedial mechanisms in case of compliance violations (i.e., by means of adaptation).

2.3 Workflows

Workflow is a technology for realizing of inter-/intra-enterprise (business) processes. There are various application areas of workflow technology; besides business workflows there are scientific workflows, grid workflows, integration logic on the middleware, systems management, ETL, and so on. The business processes act in heterogeneous, distributed and dynamic environments so adaptability, scalability, performance are required. Scientific workflows require especially reliability and flexibility since scientists plan their computational steps in advance but unforeseen situations impose the need for adapting these computations systems.

Note that workflows were originally created to support document processing and addressed static well-defined processes and so became the first successful area of the business process technology.

The paradigm of processes description has an imperative nature: graph-based/ block-based. Mainly the process is modeled as follows. There is a set of *activities* that depicts the logical steps and may be executed automatically through some applications or functions (provided by content management systems) or may require an interaction with a user. In the second case so-called *work items* are generated and correspond to activities, which are then gathered into the *work lists*. The means used to assign the work items to appropriate personnel category is called *staff assignment*. Staff assignment references a staff query, e.g. role in an organizational model instead of concrete person, department or position. This allows for better adaptability to the personnel changes in the organization. *Staff resolution* is performed during runtime and realizes the actual mapping of an activity to an appropriate person.

Workflow constructs allow to implement business process aspects like logical decision points, sequential as wells as parallel work routs, as well as managing of exceptional situations. This is realized by the means of control flow constructs of a workflow language. The business rules (complex transition conditions) specify in reusable manner the way to process the workflow specific data.

Each activity is assigned a software program or tool that supports the execution of the activity; these are the so-called activity implementations. An activity implementation can be started explicitly by a human being in charge of the corresponding work item, or implicitly in case of automatic activities.

In the next subsections we give a description of key workflow dimensions as well as address the classifications of Workflows.

2.3.1 Key Workflow Aspects/Dimensions/Perspectives

The authors of [23] outline 5 key workflow aspects or perspectives: functional, informational, organizational and operational. These are accepted as crucial workflow characteristics. They however differ from other existing classifications of workflow dimensions.

The ***Functional aspect*** of a workflow describes the functional attributes of the processes like input/output and preconditions in order to describe its functionality. The ***behavioral aspect*** specified the control flow constructs of a workflow language. The direct support of the workflow patterns (see section 2.3.4.2) is a metric of the “expressive power” of a workflow language. The ***informational aspect*** describes the data and data flow constructs of a workflow language. Constructs as parameters and variables as well as definitions and data passing are available in this perspective. The ***organizational aspect*** focuses on the properties of a workflow language regarding the capabilities to depict the organizational structure of the company and the questions how the mapping of the activities to users is realized. The ***operational aspect*** deals with the realization of a workflow application, properties like possible invocation methods and styles are included in this perspective (e.g. Java Objects, WS-RF, WebServices).

In [24] three dimensions of a workflow are introduced: the "what" dimension, the "who" dimension and the "with" dimension. The "what" dimension specifies the business logic through the control and data flow. The "who" dimension maps the organizational structure of the company in terms of roles and humans and to activities in the control flow. The "which" dimension is responsible for defining of IT infrastructure, which supports the process execution, in particular activities are assigned software programs, services and other computing resources that are executed during run time.

2.3.2 Workflow Classification

One can differentiate between the workflow systems according to the nature of the modeled processes. Crucial features of a business process are:

- Business value, which means importance of a single execution of the process to the process owner.
- Amount of required repetitions of process execution

Using these two criteria leads to the following classification (see also [24]):

- ***Collaborative Workflows*** with high business value and low amount of repetitions needed. Such workflows are used to model the complex processes that realize some specific tasks and are of great importance for the company. Examples of collaborative workflows are creating technical documentation for some product, Brand Management and others.
- ***Production Workflows*** realize structured processes of high business value and high amount of execution repetitions and usually compose heterogeneous information systems. The processes supported by production workflows implement the actual core business of the company. Therefore they directly influence the company's wellbeing and competitiveness, and therefore posses such characteristics like reliability, security, scalability, availability and performance. Examples are trip reservation, claim handling in insurance companies or loan handling in banks.
- ***Ad-Hoc Workflows*** have a property of both low business value and low repetition. The structure of the underlying processes is not defined – the behavior of users defines the process model "on the fly". Example of ad-hoc workflows is "for your information" routing. This kind of workflows involves human interaction in both controlling and execution of the processes.
- ***Administrative Workflows*** implement well-structured processes with high repetitions and low business value for the company. These processes comprise simple activity sequences and are easy to automate. Contrary to the production workflows this kind of workflows combines few information systems and doesn't realize the core businesses of the company. Altogether that implies simplified requirements on administrative workflows. Example of administrative workflow is travel expense processing.

Another criterion for categorization of workflow systems, orthogonal to the previous ones, is the degree of automation of a workflow. The authors of [22] differentiate between **document-oriented workflow**, which are mainly built out of manual tasks that manipulate unstructured or semi-structured data objects, and **process-oriented workflows**, which are mainly constructed from automated tasks which manipulate structured or semi-structured data.

2.3.3 Workflow Lifecycle

Workflow management system vendors could not agree on a life cycle model for workflows and hence there is no standard life cycle definition. However, there is a life cycle commonly supported by almost all workflow execution environments. The workflow lifecycle consists of the following phases: Modeling, Deployment, Execution, Monitoring and Analysis. Modeling defines the structure of control/data flow of the process. Deployment of processes is used to configure the process models for a particular execution environment. After deployment, the workflow is ready to be executed. In the execution phase, workflows processes are executed in the workflow engine. Thereby, the workflow engine delegates tasks to both humans and automated applications. During execution, the workflow engine writes execution logs to an audit trail. Audit data is exploited during monitoring and analysis. Monitoring gathers information on still running workflow instances, while analysis copes with post-mortem analysis on already finished instances.

2.3.4 Workflow Description Languages

There is a variety of possible classifications of workflow languages. The flexibility is one of the most important aspects for creating of agile processes; therefore in this document we address mainly the flexibility aspect. This section is but still doesn't represent an exhaustive comparison and classification of the approaches, but references the most important ones.

2.3.4.1 Imperative vs. Declarative Workflows

The most of workflow languages allow process modeling in a procedural way: all possible execution paths are modeled by means of control flow and data flow constructs. The authors of [25] argue that the procedural modeling approach tends to overspecify process models trying to predict all possible execution scenarios. The workflows of imperative nature are often not flexible enough and provide only insufficient support for ad hoc changes (migrating of the process instances to another model) well as evolutionary changes. In spite of this, the procedural workflow languages are very well accepted for practical applications and by industry and represent a trade off between the desired performance of process execution and flexibility.

There is a set of approaches provide special primitives for realizing of structural model changes and mapping those changes to the instances. This kind of strategies is also called meta-model approaches. Examples here for are ADEPTflex [29], WASA [135]. The authors of [84] mention that such approaches don't propose general resource management strategy and so don't preserve appearance of the conflict situations, which e.g. might be caused by rearrangement of resources. ADEPT is an adaptive workflow system, based on the calculus ADEPTflex, which allows controlled changes of the models during the execution. User may insert, delete or move tasks on the running process instance. Doing so the user has to have modeling expertise, that is one of disadvantages of ADEPT. The approach proposed in WASA System allows support for dynamic changes only on the instance level and focuses mainly on scientific applications.

Another category of approaches dealing with adaptability concentrate on providing of open points in the process models. The open points define some tasks in general abstract way, which can be modeled by user later at some moment of execution, or provide multiple alternative paths of the model. The examples of the systems implementing this approach are MELMAC [136], MOBILE [24] and ObjectFlow [137]. As [84] mentions this kind approaches is not flexible enough, since the open points are predefined and fixed in the process model.

The synthesized approaches separate the declaration of business tasks from their implementation. These abstract tasks are then dynamically bent to applications, resources, human participants etc (example is presented in [84]).

Another approach to describing process models and executing them in a flexible manner is enabled by the so-called case handling systems address the flexibility by focusing on the whole process instances (case) and not on single tasks. Both the flexibility and its limitations of case handling systems lie in the fact that the users are able to influence the process instance in sense of controlled deviation. A user can navigate forwards and backwards on the traditional process model (by opening re-doing or skipping single tasks). The case handling systems are a good alternative if there are too many exceptions to be modeled; it keeps the process model simple.

There is a number of other imperative approaches which address the flexibility issues, for example worklets [28], aspect-oriented languages e.g. [138]; [30] introduces the ReFFlow approach using parameterized processes for web service compositions. We will not discuss these in details but reference to the comparison section in [30].

Declarative workflows have the same understanding of activities as imperative workflow languages. Activities are here logical work units. The execution sequence of the activities in declarative workflows is defined through constraints, which implement the business rules/policies that have to be fulfilled. The constraints set the limits on the process execution paths rather than prescribing a strict execution order. [26] proposed a declarative workflow language called DecSerFlow. It specifies the constraints with the help of Linear Temporal Logic Expressions (LTL). The LTL formula can be transformed into a non-deterministic Buchi automata, which allows model checking [26]. The declarative approaches claim to support such kinds of flexibility as defer (postponing of the decision), change (ad-hoc and evolutionary), deviation (decision to ignore the model, e.g. skip, redo or swap).

2.3.4.2 Workflow Patterns

The work of [22] specified a set of patterns, which describe the control flow scenarios that may be typically required for modeling of processes. The patterns can serve as criteria for evaluation of existing workflow languages and their expressiveness. The proposed set of workflow patterns is not claimed to be a complete enumeration of possible scenarios. Besides some of the patterns appear the in real-world processes less frequently than others. This fact should be taken into account in the evaluation of workflow languages.

The work of [22] gives an evaluation of workflow languages and applications according to their support of workflow patterns (it suggests products, languages and notations such as Staffware, WebSphere, FLOWer, COSA, IPlanet, SAP Workflow, FileNet, WebSpere BPEL, Oracle BPEL, BPMN, XPDL, UML Ads, EPC).

Besides the workflow patterns there are other kinds of patterns available and new ones are currently being developed, for example data patterns [31], resource patterns [32], exception handling patterns [33].

2.4 Reusability of Business Processes

Modeling and expressing similar business processes by a party causes lots of redundancy of expression and storage, which also reduces the efficiency of the software development. Reusability of business processes is about the ability to develop reusable process fragments once and use them multiple times within the same process or across multiple different processes. If a process fragment needs to be changed it is not required to go through all usages of that process fragment to apply the change. Writing process fragments that can be reused in different places is desired practice especially in case of complex and large business processes.

Reuse is not a brand new concept in information technology community. It has been studied in software engineering, where the reuse is the process of creating software systems from existing software rather than building software systems from scratch. [154] provides a survey of software reuse.

In this setting, we will provide an overview of existing works of reuse in business process. It is about the ability to write process fragments once and use them multiple times within the same process or across multiple processes. A process fragment can be understood as a group of connected process elements that bear high potential reusability in modeling new business processes. If a process fragment needs to be changed it is not required to go through all usages of that process fragment to apply the change. Designing process fragments that can be reused in different places is desired practice especially in case of complex and large business processes. The application of reuse in business process modeling contributes to increase the quality of business processes and the productivity of business process modeling.

Reuse of business process has not be studied intensively in the area of BPM; nevertheless we classify the few existing approaches in three main classes and we review them:

1. Reuse in business process modeling: business process modeling caters for design of processes for reuse and composing new processes from reused process fragments. Both aspects are to be addressed in a lifecycle model for process fragments.
2. Semantics in business reuse: to increase the level of reusability of business processes, their semantics must be generalized and made understandable in various usage contexts.
3. Business process patterns: cater for reuse of proven process skeletons that can be customized and applied within various application domains.

In the following, we will investigate these three classes into some more detail.

2.4.1 Reuse in Business Process Modelling

Identifying and designing of reusable process fragments is not trivial. It requires a comprehensive understanding of the business domains of the organization, while keeping the current and future business goals and strategies in mind. By studying the state of the art, we have noticed that the lifecycle models proposed either by the industry or by academic focus solely on the whole picture of business process management and are in general very abstract.

Although these models provide a solid foundation for BPM lifecycle management, there is no proposed lifecycle model for using process fragments in business process modeling. In [109] the authors introduce such a lifecycle model, which (i) guides the business user in understanding and adopting the concepts of using process fragment in business process modeling; (ii) guides the development of business process modeling tool and business process repository that support reusing process fragment in business process modeling. However, the ramifications of this lifecycle model, including its dynamic behaviors and linkage to agile service networks largely remains an open issue.

In [21] the authors suggest that business process should be regarded as a kind of knowledge, and reuse of this knowledge should cover a much wider range. A business knowledge reuse framework is introduced that revolves around the notion of process components. The authors define a process component as a unit for process knowledge management and reuse. The ontology of process component is expatriated in detail. Process component model is a key factor in realizing process knowledge reuse, fostering organizational learning.

In [18] the authors illustrate the benefits of using their formal model in the first phase of the process engineering chain, namely business process modeling. Designing a new process model is a highly complex, time consuming and error prone task. To overcome this problem, the authors present a framework for supporting business users in the modeling task by reusing existing business process artifacts during modeling. This facilitates the task of modeling business processes in two ways:

- It improves the quality of the models through reuse of established and optimized artifacts;

- It reduces the process modeling time by avoiding modeling the same business process or part of it multiple times.

Traditionally, enterprise applications are built as point solutions with context-specific built-in assumptions hard-coded in their implementation. The Enterprise Application Integration (EAI) discipline deals with the mechanism for integrating such isolated applications into a consistent whole. By modeling an enterprise application as a 3-tuple comprising of its data, service and process models, EAI problems can be visualized as view-integration problem over data, service and process models. The work in [2] presents a pragmatic approach to analyze the process model of an existing application with respect to the process model of desired application to identify and mitigate the conflicts in the built-in assumptions of two process models. A formal technique to analyze the process model at various levels of granularities including a set of operators to mitigate the conflicts is proposed. The proposed approach maximizes the reusability in the context of EAI.

2.4.2 Semantic Aspects of Business Process Reuse

Business process modeling is an essential constituent of SOA application development. A formal modeling methodology has significant benefits, but related research efforts seem insufficient in the SOA context in addressing some problems in the periphery of process reuse.

The authors in [17] present an upper ontology for business process modeling, called Business Process Ontology (BPO), and propose an SOA application modeling and developing framework. The BPO provides accurate definitions of the main components of SOA modeling. The extension of BPO can be used to define business process models, describe existing services, and define the mapping between processes and services. Applying ontology to SOA Modeling can help to identify the binding information of business process and service, increase the reusability of existing business processes and services, and accelerate the development of applications.

In [16] the authors treat the reuse of the meta-model of BPs. The meta-model is the abstraction of the basic elements and rules of the process definition, and it is used to guide the process modeling of the workflow, which constitutes the core of the Process. The authors suggest borrowing techniques used in object oriented programming like Java, C# and SmallTalk. The central reuse technique in OO is inheritance, also referred to as the superclass - subclass paradigm. In particular, inheritance can be applied to allow a process to "inherit" and thereby reuse the common definition of the meta-model of its "super – process". Another technique suggested by the authors is the mechanism of "Templates", which was incorporated in the last versions of Java and which is present in C# and C++. The "Templates" mechanism can be applied to provide process skeletons that provide reusable business logic and contain “hot-spots” that can be customized to cater for domain-specific requirements.

2.4.3 Business Process Patterns

A *Pattern* is a general solution to a recurring problem or need. Patterns are formulated in an abstract and adaptable way, so to be reused in different scenarios.

In [107] the authors, seek to extract “patterns” (in the sense of Software Engineering) from existing business protocols standards (RosettaNet, XCBL, ebXML, IHE, OGSA, and ASAP), focusing in asynchronous interactions. The Patterns identified are: Callback, Publish, Subscribe, Polling, Request Response Agent Service, Callback Factory, Publish Subscribe Factory. The goal is to reuse those patterns in the creation and development of new Business Protocol. Also, based on these patterns, expressive power of the existing standards has been compared. Identified patterns are also useful in process design, since they provide certified solutions to important problems.

[108] proposes the use of patterns to help the software designer to model business processes. It focuses on the initial phases of the software development life cycle and has the objective of promoting reuse of the components of these phases. Business processes are considered to have a critical analysis phase,

which demands a significant portion of the development efforts. Due to the emphasis on these phases, the proposed solution is to use patterns with two objectives: to model the business processes and to provide reuse of analysis elements. For that, the 'Strategies and Patterns' methodology is complemented with new patterns, diagrams and stages in its process. Complementary, the pattern documentation structure is improved, and new directions are presented on how to use and to obtain patterns. To assess the propositions, one case study is presented and analyzed, trying to demonstrate the proper applicability of patterns in business processes.

3 Business Protocols and Business Transactions

This section will scrutinize the state of the art in business protocols and transactional business processes, describe their characteristics, and will explore open issues with regard to their realization.

3.1 Business Protocols

Today, service descriptions typically include the interface definition, the transport level properties (both specified in WSDL), and *business protocol* definitions, that is, the specification of possible message exchange sequences (conversations) that are supported by the service [3] to achieve a business goal. Business protocols are not executable, but protocols can be specified using BPEL or any of the many other formalisms developed for this purpose (e.g., [3], [4]). Internal details and complexity are hidden. Providing such descriptions only solves part of the problem.

To facilitate service development and interoperability there is the need for formal methods and software tools that allow the automated analysis of service descriptions to (i) identify which conversations can be carried out between two services, understand mismatches between protocols and, if possible, create adapters to allow interactions between incompatible services (called *compatibility analysis*), and (ii) manage service evolution, that is, understand if a new version of a service protocol is compatible with the intended clients (called *replaceability analysis*). Such a need is widely recognized and many approaches have been developed. We will discuss such approaches in this section.

3.1.1 Classification Criteria of Business Protocols

We can distinguish the following requirements in order to classify the existing work both in academia and industry:

- Academia and industry frameworks
- Protocol specifications
- Protocols reasoning such as analysis and management

Standardization efforts recognize the need for supporting the explicit description of web services functional and non-functional properties. Of most interest in the case of making explicit business protocols include BPEL, and the Web Services Conversation Language (WSCL). Documents complying with those specifications can be derived from protocols and vice-versa as other existing approaches are complementary to them.

Here we discuss the issues of business protocols dealing with the reasoning aspects of the exchanging message between the provider and the client for both functional/non functional properties of business protocols.

3.1.2 Specification of Business Protocols

Business processes in open settings typically involve complex interactions among autonomous and heterogeneous *business partners.* Conventionally, business processes are modeled as centralized flows, specifying exact steps for each participant. However, because of the exceptions and opportunities that arise in open environments, business relationships cannot be preconfigured to full detail. The key idea is to capture meaningful interactions as protocols. Protocols involve two or more roles and address specific purposes such as ordering, payment, shipping, etc. Protocols emphasize the essence of the interactions and omit local details. Such abstract protocol specifications are publishable as components and reusable in different settings. In what follows we discuss existing business protocol models in the literature.

Business processes are conventionally modeled as monolithic flows that capture the desired business logic. However, developing process flows is challenging. Because a flow specifies what its participants should do, it restricts the autonomy of its participants, thus limiting their ability to exploit opportunities or accommodate exceptions according to their business preferences. In [10] the authors provide a dual perspective where business processes are modeled as compositions of (instantiated) business protocols. Each business protocol specifies interactions among its partners; each protocol serves a unique business purpose, e.g., processing a payment or shipping an item. Thus, modularizing a monolithic business process via business protocols allows clear separation of concerns for modeling and enacting the process. In the proposed approach the protocols are compiled into local skeletal flows for each participant that can be fleshed out with local business logic as needed. Such flows are naturally distributed but can be enacted using commercial business flow engines. Thus, the protocol-based approach combines the benefit of improved modeling with simplified implementations.

In [11] the authors propose (business) protocols as components for developing business processes. A protocol is an abstract, modular, publishable specification of an interaction among different roles to be played by different participants. When instantiated with the participants' internal policies, protocols yield concrete business processes. Protocols are reusable and customizable, thus simplifying business process design. They show how protocols and their composition are theoretically founded in the π-calculus. They show how to formally construct composite protocols and derive processes by integrating protocols and policies. The formalization enables to reason about properties of protocols such as their incorrectness, compatibility, equivalence, and flexibility.

3.1.3 Reasoning Mechanisms for Business Protocols

As we mentioned previously, a protocol is the specification of possible message exchange sequences (conversations) that are supported by the service can be specified using BPEL or any of the many other formalisms developed for this purpose. Providing such descriptions only solves part of the problem. To facilitate service development and interoperability there is the need to do some kind of reasoning such as automated analysis of service descriptions by identifying which conversations can be carried out between two services, understand mismatches between protocols and, if possible, create adapters to allow interactions. We discuss below the reasoning mechanisms by researchers in this setting.

Modern e-business processes span multiple autonomous entities or business partners. Such processes therefore are based on a rich variety of interactions among software components that are independently designed and configured and which represent independent (sometimes mutually competitive) business interests. The authors in [1] propose a novel framework for thinking about processes. Simply put, *a process instantiates one or more business protocols* among designated parties. They defined a protocol as a specification of a logically related set of interactions. A protocol specifies only the key desired aspects of the interactive behavior; it leaves the details of a local implementation entirely up to those who implement the protocol. They concentrate on the semantic aspects of the interactions among business partners. They propose conceptual abstractions for protocols. Specifically, they consider (1) *refinement*: a subprotocol may satisfy the requirements of a superprotocol, but support additional properties; and (2) *aggregation*: a protocol may combine existing protocols. In support of the above, this work develops a semantic of protocols and an operational characterization of them. This supports judgments about the potential subclass-superclass relations between protocols, which are a result of protocol refinement. It also enables protocol aggregation by splicing a protocol into another protocol.

In [3],[5], the authors discuss the different ways in which the middleware can leverage protocol descriptions, and focus on the notions of protocol compatibility, equivalence, and replace-ability, relying on a extension of WSDL. They propose techniques to ascertain whether two services can interact based on their protocol definition, and, whether a service can replace another in general or when interacting with specific clients. The automata and simulation approaches are used to formalize the analysis and the management.

In [9] the authors deal with the problem of automated analysis of web service protocol compatibility and replaceability in presence of timing abstractions. It is an extension of [5]. They present a timed protocol model for services and identify different levels of compatibility and replaceability that are useful to support service development and evolution. Then, the contribution are (i) a model for service business protocols that supports rich timing constraints, (ii) a set of fine-grained protocol compatibility and replaceability classes, which new class of timed automata are proposed and (iii) a set of operators with formal foundations that can be combined for performing those types of analysis. The results achieved are a framework and a tool that can support development and binding of services with timing properties.

3.1.4 Protocol Evolution

Dynamism reflects changes in the business protocols requirements; modern businesses face intense pressure and must repeatedly reconfigure themselves in order to thrive, if not to survive. Supporting dynamism implies supporting process adaptability. As changes in requirements are routine, an elegant way of handling such changes is vital. Thus, dynamism poses a difficult challenge. Hereafter, we address the business protocol evolution found in the literature.

[13] aims to contribute to the aspect of adaptivity of business protocol support by collaborating partners. This aspect is a part of a broader issue of adaptivity in the business collaboration domain. In the context of B2B collaborations adaptivity has several flavours: adaptivity of the business models to different business requirements and, adaptivity of business protocols in response to business models' changes, adaptivity of the partners' end-point services to the changes in the business protocols descriptions. In addition, the business protocols should be adaptive to the changes of the partners enacting the roles defined in the protocols (both choreography and orchestration should support this).

One of the issues of autonomous Web services is that precise business protocols across systems are not always predefined. A potential solution to this problem is to dynamically generate business protocols by matching external interface definitions and correlating information exposed by each system. The current Web services have a feature called a portType (WSDL) that describes an external interface of a system. In [14] the authors propose to add a new concept called "behavior pattern" to the portType, on the basis of which an algorithm can dynamically generate business protocols. The algorithm compares the portTypes of the system under consideration and the associated system, verifies the possibility of their interaction, and automatically reduces the behavior patterns within executable range. The work also evaluates this algorithm by applying it to some use cases and shows that the method provides some useful, yet early, results for realization of autonomous Web services.

[12] addresses process adaptability studying a novel application of business protocols, especially that of protocol composition, as introduced in OWL-P (OWL for Protocols). Through a realistic business scenario involving auto-insurance claim processing, this work demonstrates how a wide range of

adaptations can be handled naturally and systematically via protocol composition. The illustrated adaptations have been evaluated with a prototype.

As the supply of Web services grows, it becomes critically important to understand the business protocols that provide clients with the information on how to interact with services. In dynamic Web services environments, service providers need to constantly refine their business protocols in order to reflect the constraints and opportunities proposed by new applications, new business strategies, and new laws, or fix the problems found in the protocol definition. However, the effective management of such a protocol evolution raises challenging problems: one of the most challenging issues is to handle ongoing instances started with the old protocol when their protocols are changed.

In [20], the authors present a framework that supports service administrators in managing business protocol evolution by providing several features, such as a set of change operators allowing modifications of protocols and two types of change impact analyses automatically determining which ongoing instances can be migrated to the new version of a protocol. They also implemented a database-backed GUI tool to manage the change process as an extension of their existing system.

3.2 Business-aware Transactions

A business-aware transaction is driven by business needs and defined as an atomic business process describing a trading interaction between possibly multiple parties that strive to accomplish an explicitly shared business objective [103]. This shared business objective extends over a possibly long period of time and is terminated successfully only upon recognition of the agreed conclusions between the interacting parties. A business transaction is driven by well-defined business tasks and events that directly or indirectly contribute to generating economic value, such as processing and paying an insurance claim, and has also an associated number of parameters that represent security and timing requirements.

A business transaction always either succeeds or fails with respect to the business task (function) that initiated it and governs it throughout its execution. If a business transaction completes successfully then each participant will have made consistent state changes, which, in aggregate, reflect the desired outcome of the multi-party business interaction.

A business transaction is made up of a requesting (initiating) business activity performed by an initiating partner (party) and a responding business activity performed by the responding business partner. The initiating business activity sends a business document to a responding business activity that may return a business signal (signifying the completion of an activity) and possibly a business document as the last responding message. A transaction is associated with an SLA that describes the agreed upon QoS requirements and usually outlines what each party can do in the event the intended actions are not carried out (e.g., promised services not rendered, services rendered but payment not issued).

3.2.1 Characteristics of Business Transactions

Conventional approaches to business transactions, such as Open EDI, the UN/CFACT Modeling Methodology (UMM) and ebXML [127], focus on the documents exchanged between partners such as invoices, purchase orders, and ship notices— possibly described in the Universal Business Language (UBL) [128].

Initially, support for business transactions within service oriented computing has however been driven by database and distributed transaction processing approaches and technologies based on resource management like JTA. First attempts build transactional capabilities directly into the business process language. Most notably, BPEL provides compensation mechanisms that are grounded on the notion of scopes (activity nesting). Although this is offers some support, [155] notes that the disadvantage is that in such event business process logic and transaction logic are mixed, process developers need to have intimate knowledge of all the transactional implications and, moreover, BPEL lacks support for isolation of results.

For this reason there has been extensive work on providing the above described transactional models. WS-Transaction (http://www.oasis-open.org/specs/index.php#wstransactionv1.1) defines WS-AtomicTransaction and WS Business Activity protocols for transaction processing on top of the WS-Coordination specification. These identify two types of transaction: atomic transactions and long-running transactions respectively. Atomic transactions conform to the classical ACID properties originating from the database transaction field; and can be nested in a close nesting model (performed all together or not at all). In contrast, long running business transactions are more relaxed in nature. They essentially constitute groups of atomic related transactions in an open-nested style. In such an open nested style transaction each of the atomic transactions represents a milestone signalling partial completion of the overall transaction. Participants can decide independently whether or not to commit results of these transactions, and as such the outcome of the transaction need not be the same for all participants. Since results are committed without knowing the final result of the overall transaction, these results are no longer isolated. Therefore, to ensure that the business process remains in a consistent state either automatic, forward or backward recovery can be performed.

In a WS-Coordination protocol there is a coordinator and a set of participants involved in a distributed activity. During execution the coordinator passes a so-called coordination context to the participants involved to keep them informed about the progress. WS-Transaction applies these concepts to the area of transactions. WS-AtomicTransaction, defines a two-phase commit and completion protocol. The coordinator ensures that registered participants reach a commit or abort decision for a transaction, and ensures that all participants are informed of the final result. WS-BusinessActivity provides two protocols for long running transactions. The outcome of a business activity can be atomic in nature (closed nested transaction) or have a mixed outcome (in which case some participants may commit results while others have to undo/compensate activities). A matter not considered in these works is how these protocols relate to other business process concerns.

In contrast to Web service transactions, which are driven by purely technical requirements such as coordination, data consistency, recovery, and so on, [103] motivates the need for using transactions that mimic real business exchanges, also incorporating the aforementioned conventional approaches, and presents an overview of several technologies and protocols that may support a business transaction framework [134] in service oriented environments. It perceives a business process as a composition of business transactions. Each of these transactions represents a consistent change in the state of the business. Additionally, [103] defines several criteria of business atomicity like non-repudiation, conversation, payment, goods and certified delivery. From a compliance point of view combinations of these criteria are interesting, as it allows compliance goals to be expressed in abstract terms that can then be made more concrete into several types of requirement.

3.2.2 Overview of Related Work

Automated business transactions are a new category of research, wider than historical data-centric local, distributed of federated transactions. This type of transactions borrow from core transactional technology, particularly the concept of an open nested transactions and multi-phase distributed outcomes (two-phase commit in conventional database/ messaging transactions).

Usage of the above solutions typically concerns layering WS-Transaction protocols on top of BPEL processes. For example, [106] suggests usage of WS-Policy assertions to declaratively attach transactional coordination constructs to BPEL partner links and BPEL scopes. During execution a generic transaction service is engaged to enforce these semantics.

Somewhat similar, [155] describes an aspect-oriented based approach to integrate a generic transaction service into BPEL. This transaction service then handles the coordination and ensures that the coordination context is included in the exchanged messages. This allows the integration of a wide variety of transaction models into BPEL without affecting its semantics; where the only limitation is the level of sophistication of the used transaction service. The difference with [106] is that in [155] attachment of transactional requirements is done in externalized scripts rather than inside the WS-BPEL process definition. This allows for a more manageable solution to both process definition and transactional requirements. This is particularly of interest as transactional requirements typically only constitute a small part of the compliance requirements applicable to business processes.

[156] proposes an initial framework in which higher-level transaction requirements are mapped onto (combinations of) lower level, basic transactions models. It would be interesting to see if this can be combined with [155] for example to construct a transaction service, which supports high level specification of transaction requirements internally supported by appropriate low level transaction protocols.

Research in the business transactions area is also related to the creation of meta-models for Web service transaction models as for example reported in [129]. In [129] a meta-modeling approach to transaction management is proposed, however, this approach focuses on the modeling and representation of transaction models driven purely from database technology perspective without taking into account business and workflow requirements.

Of other particular interest is the work on SLAs reported in [130]. Here, the authors define a template-based approach that enables automated service provisioning. This provisioning can be guided by the WS-Agreement [131] protocol. [133] proposes to enclose a transactional quality of service (TxQoS) specification in a service level agreement (SLA) to infuse transactional semantics into e- contracts for contract-driven service-oriented processes. A contract structure and a TxQoS framework are presented which enables reliability to be guaranteed both technically, via transaction management, and legally via contracts. Work reported in [132] is quite relevant as it describes many non-functional properties applicable for Web services that are also of importance for business transactions.

4 Fundamentals of Business Process Management

BPM is the successor of Business Process Integration (BPI). BPI *refers to* the ability to define a commonly acceptable business process model that specifies the sequence, hierarchy, events, execution logic and information movement between systems residing in the same enterprise (viz. EAI) or systems residing in multiple interconnected enterprises [91]. BPI is an integration solution that provides enterprises with end-to-end visibility and control over the contributing parts of a multi-step information request or transaction, which include people, customers, partners, applications, and databases. For instance, this might include all the steps in an order management, inventory management or fulfilment process.

The primary problem with business process integration lies in how a business process embedded in one application is being bridged into the process of another. The business processes linked together are described in terms of activities or workflows and bring human actors as a distinguishing element of the solution.

BPI solutions allow enterprises to take advantage of systems that are already in place by automating and managing the business processes that span these systems. With BPI, enterprises can preserve major investments in legacy systems thereby avoiding the expense of having to write additional code to replicate existing functionality.

Figure 2 illustrates a typical application of a process-integration workflow involving an order management process that ranges across organizational borders. This typical business process is shown to span multiple functional areas within an enterprise and even between enterprises. The figure shows how processes specified at the business flow-level map to corresponding enterprise information systems (enterprise services in Figure 2) at the information-flow level. The figure also shows a fork point involving the activities *ship products* and *bill customer* that are executed in parallel and a join point where these activities converge to trigger the activity issue payment. At the information flow level these two activities are shown using solid lines to distinguish them from other activities (and message exchanges) represented as dashed lines.

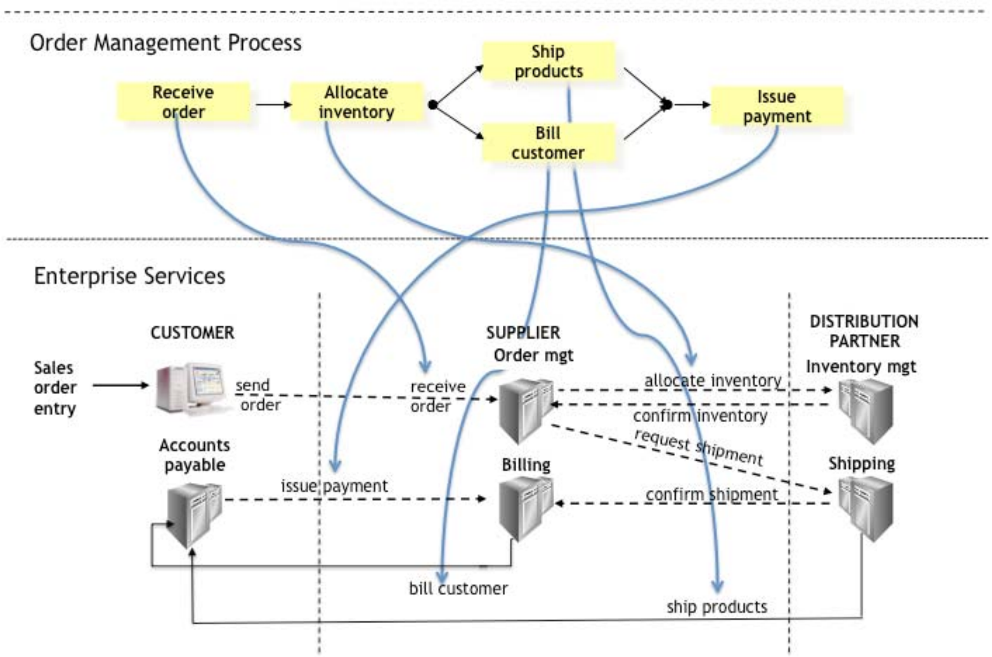

Figure 2 Example of business process integration workflow [102]

Business processes can range from being straightforward and fully automated to complex and intricate processes that require human input and intervention. The rules that govern how, when, and which users need to interact with the business process, need to be clearly identified and incorporated in the context of a business process. Thus in addition to extending critical business processes outside the enterprise firewall, modern BPI tools provide the ability to intercede at the human management level to adjust or refine or optimize business processes. This allows alerts to business processes that have gone out of boundaries, based on key performance indicators and indices used to set their tolerances. In this way enterprises can respond quickly and effectively to changing business conditions without time-consuming and expensive programming activities.

The extension of BPI with lifecycle management aspects is commonly referred to as *Business Process Management*. In the past few years, BPM evolved from its early roots in workflow to a more comprehensive system that offers graphical process design, process execution, and process monitoring and reporting capabilities for human-centric processes. In that respect BPM is more than traditional workflow as it adds conceptual innovations and technology from EAI and Business-to-Business integration and re-implements it on an e-Business infrastructure based on Web and XML standards.

BPM is a commitment to expressing, understanding, representing and managing a business (or the portion of business to which it is applied) in terms of a collection of business processes that are responsive to a business environment of internal or external events [96]. The objective of BPM is to manage the lifecycle of a process starting from business goals over process definition, through deployment, execution, measurement, analysis, change, and redeployment. The term management of business processes includes process analysis, process definition and redefinition, resource allocation, scheduling, measurement of process quality and efficiency, and process optimization. Process optimization includes collection and rendering of real-time measures (monitoring), interpretation of past measures (process analysis and mining) and strategic measures (performance management), and their correlation as the basis for process improvement and innovation.

BPM is growing rapidly as a discipline and technology for modeling, optimizing, and automating business processes. BPM is user-centric. For systems developers, BPM represents a new way to implement business solutions, emphasizing less programming and greater business involvement. For business users, BPM represents a way to participate in shaping solutions to fundamental challenges, working cooperatively with IT to improve business's performance. The result is a solution driven top-down by the business process, instead of bottom-up based on back-end enterprise information systems.

A BPM solution is a graphical productivity tool for modeling, integrating, monitoring, and optimizing process flows of all sizes, crossing any application, company boundary, or human interaction. BPM is driven primarily by the common desire to continuously improve business processes, integrate supply chains, as well as internal enterprise functions, without the need for even more custom software development. BPM codifies value-driven processes and institutionalizes their execution within and between enterprises [97], [98]. This implies that BPM tools can help analyze, define and enforce process standardization.

4.1 Phases of the BPM Lifecycle

BPM products must support all phases of the entire BPM lifecycle, as shown in Figure 3, assuming minimum custom code. This requires that BPM products have separate runtime engines; require validation, testing, monitoring and analysis tools; and facilitate the creation of new applications.

Figure 3 The Cycle of Business Process Management [102]

BPM software suites (BPMS) provide an integrated set of tools to model, design, simulate and deploy business processes and process-based applications, delivering greater degrees of process management delivery [96]. BPMS present a "closed loop" system for achieving business performance improvement, offering a set of integrated tools that support designing, measuring, monitoring, analyzing, optimizing, and continuously improving business processes.

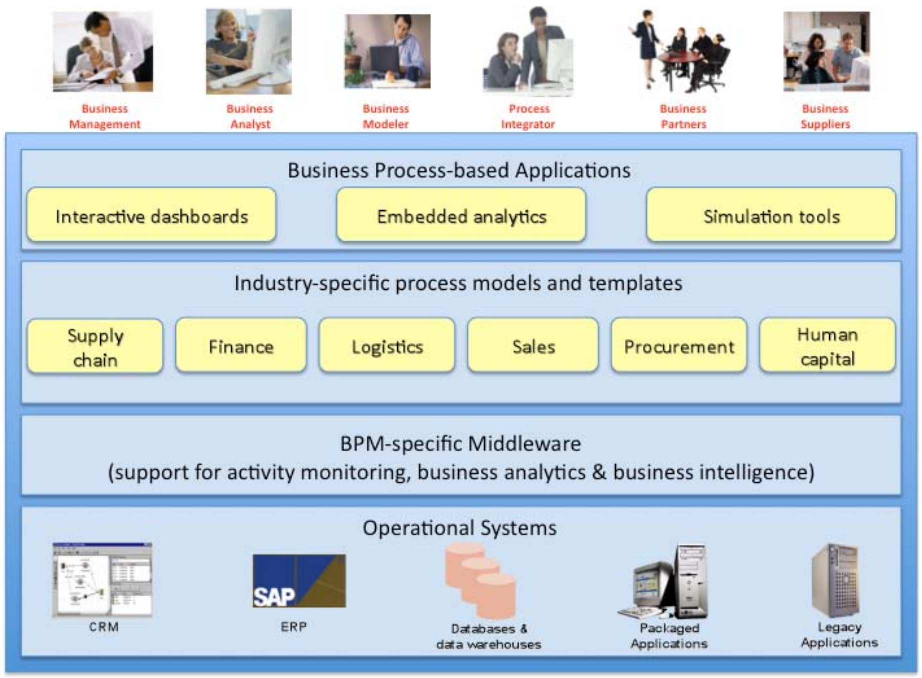

Figure 4 Typical parts of a BPM suite [102]

BPM suites coordinate tasks and synchronize data across existing systems, see Figure 4. They also help coordinate human process activities, streamlining tasks, triggers, and timelines related to a business process, and assuring they are completed as defined by a process model. A BPM suite makes processes more efficient, compliant, agile, and visible by ensuring that every process step is explicitly defined, monitored over time, and optimized for maximum productivity. A true BPMS enables business users to:

- Model and simulate all interaction patterns between workers, systems and information sources to create shared understanding about how to optimize business processes and results.
- Coordinate and manage the handoff of work across boundaries.
- Provide real-time feedback to business managers about work-in-progress to support in-line business process adjustments.
- Monitor process outcomes to performance targets, and continuously refine and adjust process flows and rules.

Based on optimal, timely and accurate business process data, business managers and analysts can take quicker action and make more competent decisions.

A comprehensive BPMS offers the following capabilities [91]:

1. *Business Process Modeling*: Process models are needed to help business managers and analysts understand actual processes and enable them, by visualization and simulation, to propose improvements. Business process modeling tools provide a shared environment for the capture, design and simulation of business processes by business analysts, managers, architects and other IT professionals. Process models are generally shown in graphical form for defining or building a business process. The key elements of a process model are individual activities performed, the events that trigger actions, the ordering of activities, the business rules used to

support decision making and execution flow, as well as exception handling and error handling mechanisms. Modern business process modeling tools include business process analysis functionality of capturing, designing, and modifying business processes and their properties, resource requirements, such as definition and selective enforcement of process standards. They also facilitate the expression of business process views at different levels of abstraction depending on authorization, functional responsibility and the level of detail desired. Process modeling is a modeling-only environment, not an execution environment. It can however support simulation. To support simulation, the models must also embrace characteristics such as skills, availability and costs of the people, and other resources that perform the process.

2. *Business Process Integration*: Connecting the process elements so that they can seamlessly exchange information to achieve business goals. For applications this means using APIs and messaging. For people this means creating a workspace on the desktop or fulfilling their part of the process.
3. *Business Process Execution*: Once the design and modeling exercise is accomplished, the process is deployed and executed within a BPM execution engine. The BPM execution engine executes process instances by delegating work to humans and automated applications as specified in the process model. An important aspect in business process execution is runtime process adaptability. Processes have to be flexible in their ability to react to changes in their environment. Such changes may be dynamics of organizational models, upcoming of better and cheaper services, business rules, compliance requirements, and so on. The goal of runtime adaptability is to change the process while it is running, without having to remodel and redeploy the process, which is in an arduous endeavor.
4. *Business Process Analysis, Monitoring and Auditing*: This involves providing graphical administrative tools that illustrate processes that are in progress, processes that are completed, and integrate business metrics and key performance indicators with process descriptions. Audit trails and process history/reporting information is automatically maintained and available for further use. Business process analysis and monitoring tools provide a wide-angle view of the time and resources consumed by enterprise-wide processes. Analytical tools guide business process improvement and deployment. Graphical reports can be also produced to check the status of all running and finished processes. These tools include facilities to query the state of live business processes and to intervene to resolve exceptions if required. For example, they can check for processes awaiting further inputs in order to complete execution, such as a process waiting for new inventory to come in. This process can then be observed and appropriately monitored.
5. *Business Process Measurement*: Managing processes first requires aggregating process data in business-oriented metrics such as key performance indicators and balanced scorecards. If the process is “out of bounds” or service level agreements are not being met, the next step is to optimize it by reconfiguring resources or modifying business rules – dynamically and “on the fly.” The ability to capture the definition of familiar business metrics and relate them to computational measurements is an essential part of BPM [96]. Business metrics definitions have an impact on which computational measurements are made. The distinction between business metrics and raw computational measurements is essential. For example, expected time-to-completion of a business transaction is of immense interest to the business analysts and management. Similarly, mean queue times, mean activity service times and most probable path to completion are too technical and too detailed for business analysts and management, however, they are essential for IT developers.
6. *Business Process Optimization*: Optimization means process improvement, which should be an ongoing activity. This item involves optimizing process flows of all sizes, crossing any application, company boundary and connects process design and process maintenance. For example, the BPM system should allow to detect bottlenecks, deadlocks and other inconsistencies in processes across the whole extended enterprise and should easily act on or

change processes in real-time to minimize inefficiencies - ideally, this should be done by the BPM system itself in an automatic manner.

These six tenets of BPM should not be considered as separate from each other. They work in concert and represent a cohesive set of actions that deliver BPM solutions.

4.2 BPM Roles and Responsibilities

Modern BPM highlights the importance of the role people play in mission-critical processes and empowers human actors in an organization, or supply chain, by providing information on demand, analysis tools to turn data into actionable information, and a secure space to collaborate with other members in the organization or value chain. BPM distinguishes between several roles and responsibilities (see Figure 4):

- It enables *business managers* to abstract business processes and rules from the underlying applications and infrastructure, and to change them directly and independently. BPM technologies provide tools that business managers can use to control and modify their processes. Specifically, BPM provides graphical models that make process explicit — that is, clearly expressed and readily changeable – and enables managers to control various aspects of business operations. After deployment, managers review reports about the business processes and make suggestions for their refinement.
- It supports *business process analysts* to deal with the more tactical aspects of BPM that is discovering, validating, documenting and communicating business process-related knowledge through modeling, simulating and analyzing current and future business process states. Typically, the business process analyst also does process and data analysis, makes changes to processes, and makes sure that any ramifications downstream and upstream from the process have been checked over, and feeds key performance indicators to business management.
- It allows the *business process modeler* who is aware of the business processes contained in an organization to capture them in a business model that can be used by process analysts. Ideally, the model is in a form that can be used also to automate those processes in the information system. Generally this is done using modeling tools.
- It supports *business process architects* to look at the various processes in the organization and describe their dependencies and other relations (process architecture). Business process architects work to resolve the inevitable differences that crop up between the business process analysts and business units. Their job also involves documenting the inter-relationships between processes and crafting a hierarchy of business processes, functional processes, sub-processes and process components tied to the enterprise's strategic initiatives.

It enables *business process integrators* to integrate existing and new services (within and between organizations), and end-users into the business process definition – the service composition components. The business process integrator will typically use visual composition tools to assemble together abstract service components that comprise end-to-end business processes. The business process integrator, along with the business process architect, will also be involved in establishing an approach to satisfying the security and Quality of Service (QoS) requirements of the enterprise when composing process-enabled applications [99].

4.3 Business Process Modeling

In the business process modeling phase, process models are created. Typically, processes are first modeled by business analysts using visual notations such as Event-driven Process Chains (EPC), Unified Modeling Language (UML) activity diagrams, or Business Process Modeling Notation (BPMN). This process model is a high-level specification of the business process, and does not yet contain the needed level of technical information to be executable. In the next step, the process model has to be translated to an executable workflow representation, such as a BPEL service orchestration and can then be deployed to a process engine in order to be executed.

4.3.1 Business Process Modeling

Business process modeling has a long tradition and consequently several well established tracks for how to model these concerns have been explored. The most prominent modeling notations are EPC, BPMN, and UML Activity Diagrams. All of these notations have in common that they support the specification of a process control flow by defining process tasks (also called activities) and their ordering. The control flow can include alternative paths of execution, define how exceptional situations are handled (fault handling and compensation), how the process reacts to events signaled by users and the environment (event handling), and additional rules and constraints.

Event-driven process chains (EPCs) is a popular business process modeling notation adopted and used both by industry and research. An EPC defines the control flow of the business process in terms of events and functions. Functions perform some business activity when they are triggered by events. Subsequently, they produce events as they carry out those activities. As such, the control flow is expressed in as a sequence of alternating events and functions. An XML-based serialization and exchange format is the EPC Markup Language (EP-ML) [139]. In addition to the specification of the process control flow other kinds of information may be included in an EPC. Such information usually depicts the use of resources for carrying out of functions and the interactions with the data structure of the organization. Because of their inherent distributed nature EPCs are suitable for both intra-organizational and inter-organizational processes. Moreover, given the importance of monitoring in compliance, events are a key component of business processes.

Even more generic in nature, another option is to utilize generic modeling approaches to describe business processes. A prime example of this is UML [140], which provides a standardized visual specification language for object oriented modeling. UML offers several types of diagrams that can be used in conjunction to model business processes. For example, class diagrams can be defined to express what information objects are involved and how these are related. Activity diagrams can then be specified to capture the process' control flow requirements by depicting what activities are performed, in what order and under what conditions. These activities are triggered by events as well as generate new events, which can be described using state chart diagrams. The activities can be linked to class diagram objects as operations performed on these objects. Use case diagrams can be specified to capture the interaction between a business process and the actors involved. Use case diagrams are very abstract in nature though and as such do not naturally lend themselves for associating actors with concrete activities.

Another recent process modeling notation is the Business Process Modeling Notation [141] developed by the Business Process Modeling Initiative (BPMI). This allows description of both intra- and inter-organizational business processes. By means of a wide spectrum of constructs, BPMN supports modeling of complex control flow scenarios and supports modeling of both private (internal) processes and abstract (public) processes. In order to model business entities and roles, BPMN uses "pools". It further allows combining the subsets of activities of a business process into "swim lanes" and so makes the mapping of activities to organizational units possible. Besides allowing to describe the control flow of a participant process, BPMN aims to support modeling of the collaboration of several processes. While the control flow describes behavior of each participant and specifies the ordering of the interaction-relevant activities inside of each participant (BPMN pool), the message flow "ties" the participating processes into conversations.

Another interesting research area is that of role and agent based approaches. The motivation behind such approaches is that often in process models it is unclear who is responsible for what, since many of them (such as workflow) tend to have decomposition related to function. Therefore, works like

[143] and [144] emphasize definition of resource usage. The work in [143] for example suggests to specify the foundation of a meta-model of a business process definition in terms of message exchanges between two roles. This enables the definition of decentralized processes which involve business-to-business collaborations as well as user interactions and application-to-application integration in a technology neutral way. [144] presents an autonomous agent based approach to business process modeling management in which processes are viewed as interactions among independent agents. Differently, work in the semantic web has concentrated on the informational aspects of business processes taking the idea that information is key to the business process. Such approaches start out by defining the data structure in a process. Subsequently, operations that can be applied to this structure are developed, which constitute the different process tasks. Other constructs can then be added if desired. A well known exponent of such form of business process modeling is DAML-S, which is being developed by the DAML Program [145].

A separate category of process modeling languages is formed by more formally oriented works. Formal techniques contribute to the formal verifiability of business process models (e.g. concerning deadlock). However, their usage requires expert knowledge. As such, a combination of a user friendly modeling language and an underlying formalization would be a good solution. One example is simple finite-state automata, with which processes are described as devices that maintain the state of something at a certain time and can alter this state in reaction to input as well as cause an action or output as a result of a changing state. Petri nets offer another graphical technique, and are a special form of graphs constituting of places, transitions, directed arcs and tokens. Places are connected via directed arcs to transitions and vice versa. Places contain tokens, which may represent signals, events, conditions, and so on. Transitions are fired through the presence of tokens in their in-place(s). As a result the distribution of tokens is changed. Example works include [146] and [147] using Petri Nets for workflow and service composition representation respectively.

More symbolic in nature are the techniques of formal logics and process algebras. Formal logics define processes as collections of predicates based upon which reasoning can take place to analyse characteristics of these processes. Process algebra is a formal description technique designed for complex computer systems, especially those involving communicating, concurrently executing components [148]. Many process algebras exist, such as CSS [149], CSP [150] and ACP [151]. An algebra that has received some popularity specifically in the context of web service based business processes, is pi-calculus [149]. In pi-calculus a process can be described as a collection of interacting services over communication channels. A last approach worth mentioning is REO [152]. REO is a constraint automata based approach for formal definition of foundational models for coordination and composition. It is characterized by the fact that that they can cater for the description of the behaviour of active entities that (1) are fully compositional, and (2) can express arbitrary mixes of synchronous and asynchronous behavior.

4.3.2 Executable Business Process Models

The process model created by a business analyst using a visual process modeling notation does not yet contain sufficient technical information to be executed. In the next step it is either first refined by a developer who selects services which can be used for implementing process activities, or translated to a technical representation of the service orchestration which is understood by the process engine. The standard language for specifying service orchestrations is WS-BPEL (Web Service Business Process Execution Language (or BPEL) [153]). The translation of business process models to executable process models (e.g. in BPEL) can be difficult and has been extensively investigated, e.g. in [110], [111], [112]. The resulting orchestration model in most cases has to be manually adapted adding information on data handling, exception and compensation handling. The service orchestration is finally deployed to the process engine.

BPEL is a combination of block-based and graph-based constructs. It is an XML-based language for composition of Web Services. It utilizes the process-based approach for compositions, where the process logic is separately defined from the discrete activity implementations. It is used as an exchange format for (business) process models and as an executable format. BPEL is considered an advanced orchestration model, where a process can be used as a Web Service, i.e. a process itself is exposed as a Web Service and thus can be accessed using the same mechanisms as any other Web Service thus contributing to interoperability.

Two kinds of business processes are distinguished in WS-BPEL: abstract and executable processes, although the same syntax is used for abstract processes as for executable ones with some extensions. Abstract processes describe business interactions by precisely specifying the message exchange behaviour of the parties involved without revealing their internal implementation. Importantly, in this the separation of public and private view on the business process for the purpose of either specifying a template for a business process or the visible view on the behavior of the process. Executable processes contain all the information necessary to execute a process excluding the deployment information relevant to concrete services to be used or selection criteria for their discovery during deployment or execution, done by the service bus [83] (the middleware for services). A more extensive discussion on WS-BPEL is provided in deliverable PO-JRA-2.2.1.

4.4 Business Process Execution Environment

The BPMS execution environment navigates through the process models to proactively move forward their instances of execution. In doing so, the execution environment orchestrates the instances and coordinates the human work items, rules-based automation, and system-to-system integration activities. Making a business process model executable requires BPM execution-enabling software such as an orchestration engine, integration services and a run-time environment as well as rule engines for execution of the flow between end-to-end processes and human actors. The orchestration engine oversees the course of properly sequencing process activities according to the flow definitions and rules in the process model into an end-to-end business process (within an organization), assigns work items to the appropriate human actors or groups, and ensures that both human- and systems-based activities are performed within specified timeframes. This entails multiple technical requirements, which include binding to heterogeneous systems, synchronous and asynchronous message exchange patterns, data manipulation, flow coordination, exception management, business events, long running compensating transactions, and so on.

BPM orchestration engines rely on standards support such as native support for BPEL and Web services standards. Many BPM vendors are currently expanding to include BPEL as an orchestration language achieving thus a better utilization of the services of a Service Oriented Architecture (SOA), see section 4.6. However, although BPEL is geared toward orchestrating business services out of relatively fine-grained service interfaces — much in the way that SOA uses orchestration – it is not capable of choreographing end-to-end business processes and achieving global visibility as required by sophisticated BPM tools.

Most BPMS products also interact with business rules engines to automate those aspects of business processes that involve complex decision-making. In most cases the business rules engines are not a part of the core BPMS product, but are provided as an additional capability that can be used whenever needed. Rule engines execute rules that abstract business policies and decision tables from the underlying applications, and make available more-flexible process changes. (see following section) Finally, integration services assist orchestration engines by enabling BPM to leverage advanced connectivity and transformation capabilities for business processes. These capabilities include, for instance, support for XSLT/XQuery transformation and connectivity to multiple packaged applications and legacy systems [101].

Sophisticated BPMS use BAM facilities to provide business insight with process execution thereby improving decision-making and driving process improvements. This has given rise to the concept of Business Intelligence. Business Intelligence is a broad category of applications and technologies for gathering, storing, analyzing, and providing access to data to help enterprise users make better business decisions. Business intelligence and analysis tools support analysis of data produced during process execution. The emphasis is on prediction of future behavior such as forecasting, scenario planning, optimization, which are derived on the basis of analytical processing and inferencing.

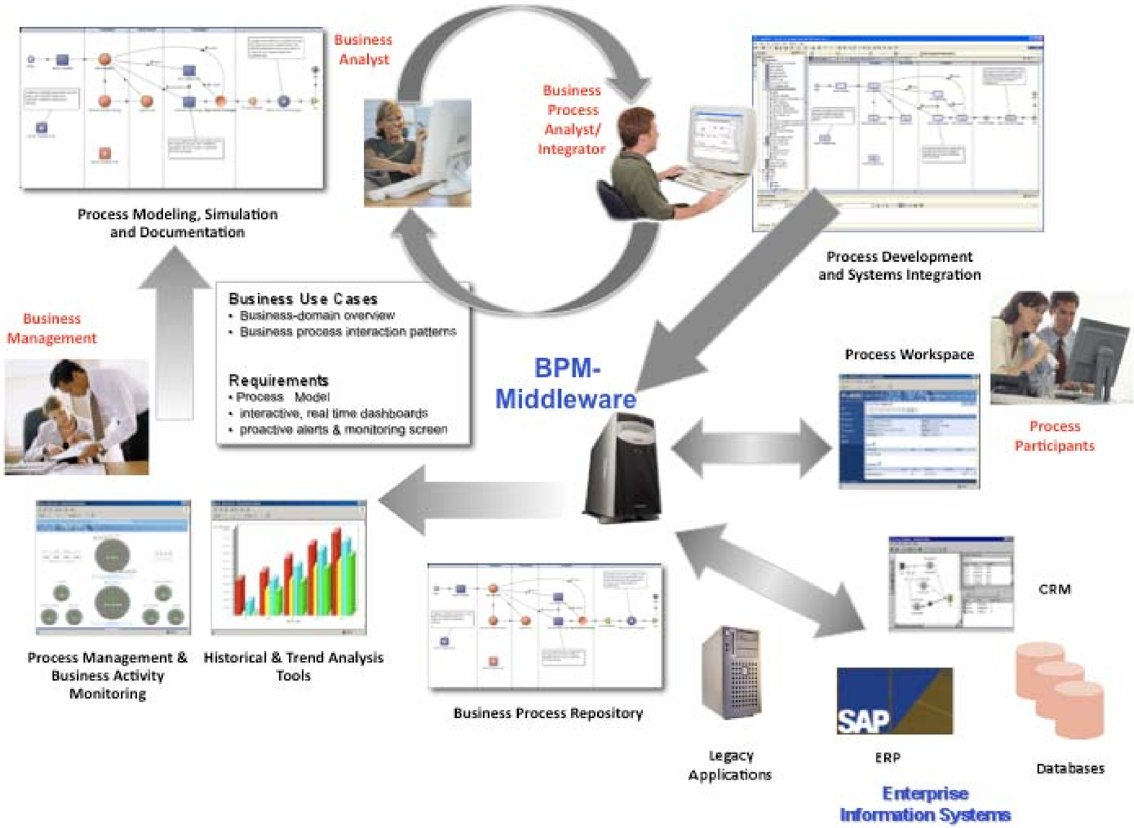

Figure 5 BPM Execution Environment [102]

Business intelligence applications include the activities of decision support systems, query and reporting, online analytical processing (OLAP), statistical analysis, forecasting, and data mining. Business intelligence software helps enterprises access vast amounts of information collected across enterprises and perform analysis so that they can mitigate risks and predict future outcomes – all without users needing advanced statistical skills – using query-and-reporting capabilities. This helps enterprises make effective us of business-related information in order to better manage, measure and optimize business performance and to do so in a repeatable manner.

Business intelligence analytic capabilities, along with data management and query-and reporting technologies, comprise analytic applications that solve key business pains, such as misalignment within the organization, lack of customer knowledge and slow response rates to market shifts. These applications bring business and industry context to the specific use of analytics for better business planning and strategy implementation. Business intelligence applications can be:

- Mission-critical and integral to an enterprise's operations or occasional to meet special requirement.
- Enterprise-wide or local to one division, department, or project.
- Initiated or driven by user demand.

The Business Intelligence infrastructure may include executive dashboards, scorecards, and other tools that make it easy for business managers to find and understand the information and proactively use it in decision-making, see bottom left part in Figure 5.

4.4.1 Runtime Process Adaptability

An important aspect in business process execution is runtime process adaptability. Processes have to be flexible in their ability to react to changes in their environment. Such changes may be dynamics of organizational models, upcoming of better and cheaper services, introduction of new business rules and regulations and so on. The goal of runtime adaptability is to change the process while it is running, without having to remodel and redeploy the process, which is in general very time-consuming.

Run time changes may happen on model level or instance level. Model level changes must be propagated to all or some of the running instances of the models changed, also called instance migration; instance changes have only an effect on corresponding instances which are to be adapted; this corresponds to the so-called non-permanent [84] or momentary [85] changes known from workflow systems. The concepts developed to enable flexibility by adaptation on the process logic level [86] can be reused in Web service workflows (WS-flows), thus control flow changes, although rarely necessary, may be supported on engine implementation level.

While runtime adaptability of conventional workflows and Web service workflows is similar, in the context of WS-flows in particular dynamic discovery and binding of services becomes important. The goal thereby is to change the implementation of a process activity at runtime. The change might be needed, because the service which was bound at design-time is no more available or has faulted, or because one wants to always select the most appropriate service considering non-functional properties such as price, response time, and security.

Implementations of process activities in BPEL are represented by WSDL port types and operations of participating Web services. Port type and operation names are statically defined during process design time. At process runtime, if one wants to bind to another service, two cases have to be distinguished: (i) the alternative service implements the same service interface, i.e. WSDL port type; (ii) the alternative service implements the same functionality but a syntactically different port type. In the first case only the implementation of a service interface (port type), the WS port, is changed. Some cases for which such adaptations would be required are, for example, the need to repair a process due to inability of the infrastructure to find an appropriate partner (or to swap an unavailable partner). [87] shows how dynamic binding of services implementations can be accomplished for BPEL processes. The BPEL interaction activities are extended by find-bind functionality, which enables searching for a service in a service registry and binding it to the process instance at runtime.

In the second case, the problem is that there are two or more services of the same functionality, and all of them advertising a different service interface. A process statically referencing by name concrete operations and port types passes up the advantage of using the functionality exposed by another provider. For this reason WS-flow languages such as BPEL are not flexible enough because they are still statically defining the WS types they utilize. Should a process owner wish to use an alternative provider he must either adapt the process models and/or instances during run time, which means that adaptability would be necessary, or one must be able to model their processes in such a way that inconsistent changes are avoided. In [88] an approach is shown which enables dynamic binding independently of the predefined port types. The approach extends BPEL interaction activities, which can specify different strategies on how the dynamic binding is to be accomplished at runtime. In particular, it is possible to pursue a semi-automatic approach where an administrator is prompted at process runtime to provide an alternative service, e.g. when the predefined service has failed, or a fully automatic approach, which uses a semantic query to find functionally equivalent services. As WSDL Web services only syntactically specify the functionality they provide, a fully automatic approach is not possible without a semantic description. In [88], for semantic description OWL-S technology is used. In [89] and [90] an extension to BPEL, BPEL4SWS, and the implementation are presented which support interaction with semantic web services.

More details on the state-of-the-art in adaptability can be found in the deliverable PO-JRA-1.2.1.

4.5 Business Activity Monitoring (BAM)

The term Business Activity Monitoring (BAM) refers to near real-time monitoring of business activities, measurement of key performance indicators (KPIs), their presentation in dashboards, and automatic and proactive notification in case of deviations. A "business activity" thereby can be implemented as a service orchestration in a BPMS, or, more general, as part of a business process consisting of a series of activities implemented across workflow systems, ERP systems and legacy applications, possibly across organizational boundaries. BAM software gathers information from these applications in form of events, aggregates and analyzes these events to compute KPIs, and reacts to deviations by notifying business users. In this deliverable, we will take stock of those concepts and techniques in BAM that are particularly relevant for process management of agile service networks. A comprehensive analysis can be found in PO-JRA-1.2.1.

The goal of BAM is to provide timely information about the status and the performance of business processes, and to proactively alert users if business rules are violated or KPI values deviate from target values. Compared to post-mortem reporting and analysis based on audit logs, BAM technology allows business users to react to changes and violations in the business environment more quickly.
In general, a BAM solution offers the following capabilities:

- Definition of business rules and KPIs which are to be monitored at process execution time
- Personalized visual dashboards which display information on the status of the processes and KPIs in near real time
- Proactive alerting in case of violations of business rules or deviations from KPI target values

4.5.1 Event Processing

Business Activity Monitoring is based on event processing. Events have to be gathered from different kinds of applications: process engines, ERP systems, databases, legacy applications. Not all of these applications support publishing of events natively, so often adapters have to be implemented which extract events from these applications. BAM tools subscribe to these events and typically use Complex Event Processing (CEP) technology, which enables them to process high volumes of underlying technical events to derive higher level business events. A rule engine calculates KPI values based on business events and sends notifications. Events and KPI values are typically stored in a data warehouse of the BAM tool, which is queried by the dashboard component for visualizing the KPI values.

While for legacy systems adapters have to be implemented for extracting events, BPM systems provide an event publishing mechanism out-of-the-box. Thereby, events are published to denote state changes of the process instances and process activities (e.g., started, halted, resumed, finished, faulted). Each process engine vendor, however, implements a different event model and auditing mechanism. Thus, a monitoring tool would have to provide adapters for each (BPEL) process engine it wants to support.

In [78] Roth et al. deal with the problem on how to extract events from a BPEL process in order to enable auditing in an interoperable way. The presented solution extends a BPEL process model definition with special auditing activities which log state changes to an external monitoring web service. The extended BPEL process does not use any proprietary elements and is BPEL standard compliant. Therefore, the extended BPEL process can run on any process engine and send events to the monitoring tool. First, the authors introduce five different strategies for auditing BPEL processes: (i) instrumentation of web service requests of the BPEL process on protocol level and (ii) on application server level, (iii) utilizing the auditing service of a process engine used for enacting the BPEL process, (iv) using probes in the operational systems that track state changes of the business process, or (v) including the auditing mechanism as a partner within the BPEL process. They employ the fifth strategy and show how to transform a BPEL model into an auditable model which can be used for process monitoring purposes. For every audited activity, a new scope is created which hosts and executes all the necessary steps for pre- and post-auditing. The benefit of the approach is that the extended BPEL process does not use any proprietary elements and is BPEL standard compliant. Therefore, the extended BPEL process can run on any process engine and send events to the monitoring tool.

4.5.2 Development of BAM Solutions

Normally, there is considerable effort involved in developing and deploying BAM solutions. Business users first specify KPIs with target values and rules when they are to be notified. IT engineers take this specification as input and create the needed IT artifacts. First one has to determine which events are needed and how they are to be extracted from existing applications. This possibly involves implementing adapters. Then, based on these events a monitor model has to be created which specifies how to calculate the KPIs based on the events, and in which cases to send alerts. Finally, dashboards views have to be configured.

In [77] Momm et al. deal with the problem on how to develop automated SOA-based business processes with integrated monitoring information for process controlling. Automated BPEL-based business processes are often developed in a top-down manner, starting with a visual notation of the process (e.g. in BPMN) and then translating the visual model into an executable BPEL process model. If the BPEL process is to be monitored, then also process metrics have to be specified during process development. The presented solution utilizes a model-driven approach to developing monitored business processes. The authors have created a metamodel, which allows modeling of process performance metrics (PPIs) based on BPMN process elements. The BPMN process model with the corresponding PPI model is transformed to a BPEL process model, which contains additional activities for publishing events needed for the calculation of the PPIs. These events are sent to an external monitoring tool by invoking its Web service interface. For measuring the duration of the activity, for example, two additional BPEL invoke activities would be inserted, before and after the activity, respectively. These activities would invoke corresponding operations on the monitoring tool. The benefit of the approach is that events needed for monitoring are automatically determined and corresponding activities for event publishing are automatically generated.

In [76] Chowdhary et al. present another model-driven approach to development of BAM solutions. The authors have created a business performance management metamodel for modeling of monitoring tasks in a platform-independent way. The Business Performance Management metamodel contains concepts such as “Metric”, “Business Event”, “Situation”, “Business Action”, etc. The user can create such models using UML. The business performance management model is transformed to two intermediate models: an observation model and a data warehouse model. The observation model captures information for the monitoring tool on how metrics are to be computed and which actions to take in certain situations. The data warehouse model deals with storage of metrics, and their visualization in dashboards. Both intermediate models can be adapted if needed. Finally, they are transformed to deployable code. The benefit of the model-driven approach to creation of monitoring solutions is that much of the code can be fully generated.

4.5.3 BAM vs. Other Forms of Process Monitoring and Analysis

The BAM approach should be differentiated from other forms of business process monitoring and analysis. [74] presents a taxonomy for workflow analysis distinguishing between two dimensions: (i) technical analysis vs. business-oriented analysis and (ii) whether the analysis involves live data or history data. One can then further distinguish whether the monitoring tasks focuses on single instances or multiple instances. Technical analysis involves, for example, system monitoring (live data) and workflow recovery in case of system failure (history data), whereas business oriented analysis copes with tracking of the individual process state (workflow monitoring) and workflow controlling which deals with the business-oriented ex-post analysis of potentially finished workflow instances. Technical monitoring also involves detection of participating services in a service orchestration that are erroneous and deliver unexpected results [79]. In this taxonomy, Business Activity Monitoring covers business oriented analysis based on both live data ("number of open trouble tickets") and history data ("average duration for resolving an incident"), and for single instances ("if a purchase order is not processed within 48 hours, send an alert") and multiple instances.

BAM solutions are similar to reporting tools and business intelligence tools when it comes to display KPIs in dashboards. The distinction is that BAM tools process events in near real time, calculate KPIs and push the results to the dashboard. BI dashboards on the other hand query a data warehouse post-mortem when requested by the user or refresh the views periodically. Depending on the refresh interval selected, BAM and BI dashboards can be similar. Often, BAM tools also use a data warehouse schema for efficient querying of metric values. Another difference is that BI tools, in addition to reporting the values of KPIs, use data mining techniques to analyze the causes KPIs not reaching target values.

In this context also, there are approaches which attempt to combine BAM and data mining for explanation of KPI values. In [75] Castellanos et al. deal with the problem on how to create a monitoring solution, which not only enables to measure KPIs, but also to understand the causes of undesired KPI values, and prediction of future values. Conventional BAM solutions let the users define business metrics and then monitor and report them at runtime. They however do not support explanation of the causes of certain metric values to the user. The authors sketch a platform which combines business activity monitoring with data mining approaches to enable more intelligent analysis of business metrics. The platform should support following functionalities: (i) providing visibility into processes which are not executed by a process engine, but run implicitly across diverse systems; (ii) enabling the business user to define KPIs in an intuitive way; (iii) enabling explanation and prediction of KPI values. In order to provide visibility into processes, the authors take the approach to model a so called abstract process which models the steps of the process in terms of events which are to be extracted from existing systems. Events are extracted by using adapters. The abstract process model is not an executable process model, but serves as an input to the monitoring tool in order to display the status of the process as events are received. In order to enable the business user to specify business metrics, a template based approach is employed. Thereby, IT engineers specify business metric templates using SQL queries over the underlying event data store. Business users instantiate the templates with concrete values, thus specifying the monitoring tasks without further support by IT engineers. Finally, for explanation and prediction of metric values, data mining techniques based on decision trees are used.

4.6 Service-enabled BPM

Service orientation utilizes services as constructs to support the rapid, low-cost and easy composition of distributed applications. Key to this concept is the service-oriented architecture, which is a logical way of creating loosely coupled, interoperable services that can be easily shared within and between enterprises, via published and discoverable interfaces [100]. Enabling loosely coupled interactions of services between service consumers and service providers allows the creation of business processes and complex applications in which service compositions are essentially business processes.

Service Oriented Architecture focuses on services which expose discrete functionality using a publish-find-bind approach. A standard way to describe service interfaces hides from users the complexity of combining software components built on the basis of heterogeneous platforms and technologies. The SOA "publish/bind/find" triangle provides the functionality needed to make services available and accessible [100]. A provider of a web service is from the business point of view an organization that realizes a piece of business logic and is responsible for publishing the service in a service registry. Publishing means providing needed descriptive information about the service: both business-related information as well as technical information. Service requestor is from business perspective the client enterprise that requires the service. Requestor searches for the required service by asking the discovery facility and then uses the received information for binding of the service.

A well-constructed SOA can empower a business environment with a flexible infrastructure and processing environment by provisioning independent, reusable automated business processes (as interacting services) and providing a robust foundation for leveraging these services [101]. Business processes form the foundation for SOA and require that multiple steps occur between physically independent yet logically dependent services. Underlying the need for flexibility in SOA is the ability to rapidly assemble new services and business processes to address business needs.

BPM is a natural complement to SOA, and a mechanism through which an organization can apply SOA to high-value business challenges. The objective is to effectively align technical initiatives with the strategic goals of the business user at every level within the organization and between organizations to achieve a comprehensive approach to real business transformation.

Both SOA and BPM can each be pursued without the other, but the two approaches in concert offer reciprocal benefits. Layering BPM on top of a solid SOA allows actions within business processes to be exposed via automated services. With BPM orchestration, the exposure of key business events and information to users at the appropriate times and in the appropriate contexts adds tremendous business value that might not otherwise be achieved with a SOA not making use of BPM. In addition, BPM helps deliver control over business processes, fostering standardization across a company or an end-to-end process chain and compliance with regulations, policies, and best practices. It also enables some services required by the business process to be outsourced to trading partners, and opens up brand-new business models in which the enterprise's own business processes can be exposed as services to new customers, both internal and external.

A layered service-enabled BPM model proposed in [101]. This model contains six abstraction layers depicting top-down development approach: domains layer, business processes, business services, infrastructure services, service realizations, operational systems. While the topmost layer combines the enterprise processes from the same business domain, the second layer comprises the business processes which are able to collaborate in order achieve business objectives. The third layer is the orchestration layer: here the business processes are decomposed into a set of smallest sub-processes which are then candidates to become business services. Business services are in turn supported by infrastructure, management and monitoring services such as those providing technical utility, for instance, logging, security, or authentication, and those that manage resources. These provide the infrastructure services often considered as part of the Enterprise Service Bus (see deliverable PO-JRA-2.3.1). The services infrastructure layer also provides technical utility services required for enabling the development, choreography, delivery, maintenance and provisioning of business services as well as capabilities that monitor, manage, and maintain QoS such as security, performance, and availability. The realization layer in an SOA identifies and characterizes a large number of components that provide service implementations in terms of operational functions and data available from resources (operational systems) such as ERP, databases, and CRM systems as well as other enterprise resources. Components that are revealed in this layer represent the building blocks for business services and processes in an SOA.

By providing services at the right service granularity and enabling composability of services in service orchestrations, SOA helps bridging the so called Business-IT gap. The Business-IT gap denotes the difficulties in aligning IT to the goals of the business and implementing and executing business processes in an efficient manner. In this context, Business-driven development deals with the top-down development of executable business processes. The first step consists of modeling a business process in a visual notation such as Event-driven Process Chain (EPC) or BPMN. This process model is created by a business analyst and does not yet contain the level of technical information to be executable. In the next step it is either first refined by an IT engineer who selects services which can be used for implementing process activities, or straight away translated to a technical representation, e.g. BPEL. This translation can often be rather difficult and has been extensively investigated, e.g. [110], [111], [112]. The resulting orchestration model in most cases has to be manually adapted adding information on data handling, exception and compensation handling. The service orchestration is finally deployed to the process engine and an enterprise service bus. Besides the process model itself several other artifacts have to be created in this process, such as monitoring information, graphical user interfaces in case of human tasks, service binding information and so on. The benefit of using SOA for the implementation of business processes is the ability to easily select and orchestrate services for implementing business functions.

5 Agile Service Networks

Today's application-oriented services cannot scale to meet the number and nature of demands already placed on them, much less a new generation of more complex applications involving several organizations that need to collaborate in a dynamic manner. Most of today's production applications are based on the assumption of the ubiquitous availability of point-to-point integration between any two interacting parties from the perspective of a single organization. One of the main reasons is the fixation of current orchestration languages (exemplified by BPEL) to describe how services can interact with each other at the message level from the perspective and under control of a single services. Moreover, the interactions are limited to uni-cast scenarios. In addition, current BPM deployments have been narrow in scope, providing improvements in specific business functions (i.e., invoicing, shipping, inventory, compliance, etc). Moreover, many current BPM systems only enable organizations to enhance their existing processes. This is extremely restrictive for applications characterized by wide-scale and complex dynamic interactions.

The next-generation of service-enabled BPM will serve as a means of developing mission-critical applications based on strategic technology capable of creating and executing cross-enterprise collaborative business processes, business-aware transactions and connecting the entire business value chains. With a process-managed business value chain, organizations can deploy, monitor and continuously update cross-enterprise functions within a mixed environment of people, content and systems. The next generation of service-enabled BPM will essentially provide much more functionality and flexibility, enabling organizations to innovate new value delivery systems that transcend the enterprise and extend to every external partner. The trend will be to move from a relatively static view of an organization to a much more dynamic, high-value one where end-to-end business process interactions and trends are examined much more closely to understand much more accurately the business dynamics. Such collaborative, complex end-to-end service interactions give raise to the concept of Agile Service Networks.

Agile Service Networks comprise large numbers of long-running, highly dynamic complex end-to-end service interactions reflecting asynchronous message flows that typically transcend several organizations and span geographical locations. The term complex end-to-end service interaction signifies a succession of automated business processes, which are involved in joint inter-company business conversations and transactions across a federation of cooperating organizations. This widens considerably the scope of service-based applications by providing the possibility of developing a whole new range of innovative service-based applications.

Agile Service Networks (ASNs) provide global (end-to-end) process visibility and monitoring, identify business goals, and determine strategies to achieve these goals. They rely upon powerful analysis capabilities to provide an effective approach for targeting business problems in areas like compliance, change management, quality improvement, and operational business health, delivering more business value and reducing risk. To achieve these capabilities they properly sequence service activities according to the flow definitions in a process collaboration model into end-to-end constellations, assign work items to the appropriate human actors or groups, and ensure that both human- and systems-based activities are performed within specified timeframes and under the right conditions. This entails multiple technical requirements, which include binding to heterogeneous systems, synchronous and asynchronous message exchange patterns, data manipulation, flow coordination, exception and change management, handling of various business events, long running business transactions, and so on. In addition, business activities that take place in ASNs need to be structured, modularized, reused coordinated and synchronized with other relevant activities within the service network in order to improve or optimize their operation.

Current practices and methods do not address the aforementioned issues. Emerging service networks require models, languages and methods for describing, analyzing, assessing, designing, and managing a firm's internal and external services and processes. New platforms and tools for business process management and adaptive capabilities over open services frameworks are needed. To address such requirements, we need to study models that aim at answering basic questions that focus on how a service network can be formed (created), depicted, and evolve. For example, we should specify and analyze the conditions under which a service network can be dynamically restructured and how one party may leave one network to join another, and so on. These conditions could drastically impact network partners' business objectives, the network topology (structure) and network-wide business processes. In addition, we need to also study, analyze and predict emerging dynamic business patterns that result in dynamically forming ASNs. This results in the need for designing and studying algorithms for emergent service network creation, optimization, evolution and destruction, and a subsequent focus on corresponding tools that help visualize, analyze and simulate the evolution of service value networks.

The remainder of this section reviews the state-of-the art in agile service networks, concentrating on key techniques and concepts for leveraging BPM and SOA.

5.1 Business Goals, SLAs, KPIs and Metrics

In order to evaluate and measure the performance of a firm within an ASN and define business objectives as part of the firm's strategic behavior, the firm identifies specific Key Performance Indicators (KPIs). KPIs are quantifiable metrics that a firm uses to measure performance in terms of meeting its strategic and operational objectives [67]. KPIs provide critical information to the organization for monitoring and predicting business performance in accordance with strategic objectives in a way that compliments financial performance. By measuring and monitoring operational efficiency, employee performance and innovation, customer satisfaction, as well as financial performance, long term strategies can be linked to short term actions.

On the other hand, SLAs are considered to define the agreed performance and quality of the service or product defined between the provider (the business, enterprise or department) and the ultimate end-user (the customer) [68]. Business process management combines information acquired from value network analysis with requirements enforced by KPIs in order to meet the agreed SLAs. A mapping of process SLAs to value network KPIs is needed due to different semantics of the KPIs on the IT and business level.

There are several studies that define, analyse and monitor KPIs ([69],[73]). In [69], a formal language for performance indicators and their relations and requirements is presented and applied to a case study in logistics. It is shown how this language can be used in informal, graphical or formal form. A software environment has been developed that supports the specification process and can be used to automatically check whether performance indicators or relations between them or certain requirements over them are satisfied in a given organisational process.

In [70], non-financial indicators (e.g., customer satisfaction, motivation, safety, quality of delivered goods, etc.) are defined that give valuable information for the effective management of an organization. In particular, a taxonomy of performance indicators is introduced, namely the balanced scorecard, that describes in a comprehensive way the business objectives. The balanced scorecard includes financial measures as well as operational measures on customer satisfaction, internal processes and the organization's innovation and improvement activities. In particular, the balanced scorecard allows managers to look at the business from the customer, internal, innovation and learning, and financial perspective.

In [71], [72], [73], performance indicators are shown to play an important role in supply chain management since they affect strategic and operational planning of an organization. In [71], an empirical analysis is performed based on literature on performance measurements and metrics in supply chain management in order to improve the value chain performance. In [73], an analysis of various performance metrics for supply chain management (SCM), used by a specific manufacturing company is performed. The paper proposes to deal with multiple metrics in SCM and shows how simulations are used in order to forecast how the values of these metrics will change once a supply chain is redesigned.

5.2 Design and Development of ASNs

There are several studies focused on creating or reconfiguring agile service networks ([36], [37], [38], [39], [40], [41], [42]). Noticeable among them is the seminal work by Verna Allee, such as the **ValueNet Works™** analysis, using the intuitive **HoloMapping™** method, which is a methodology for analyzing the dynamics of value in value networks at the operational, tactical, and strategic level. The emphasis is on visualization and qualitative methods. In the "e3value" approach, the authors present an e-business modeling approach that combines IT systems analysis with economic-based business modeling [43]. They focus on building an e-business model that specifies relationships and e-business scenarios rather than on defining values.

In addition, there is a growing need for quantitative methods. Newly deconstructed functions must be priced to generate return through market mechanisms and the deconstructed price structure should merge into the final cost and value delivered through the service system. Alternative designs of business restructuring or of business alliance formation may have to be evaluated. Dependencies among participants also influence the value finally accrued. In Caswell et al. [44] a start is made on the problem of modeling value in service systems by defining an analytical framework. The general problem statement comes from real-life scenarios such as the automotive and electronics value chains, where approaches for optimizing value, cost and information flows are open and have not been looked at yet.

Network formation by (economic) agents has been studied in the literature [45]. The objective there is to form both effective and stable networks, which in general is difficult to achieve. The definition of value used as the benefits of an agent accrued by his participation to the network, minus any costs involved in setting up the network links directly or indirectly, is a static value definition.

Collaborative networked organizations are not only motivated by their value models: as they enhance the enterprise's agility i.e. the ability to answer to structural changes quickly (client requests, technological or activity changes, supplier management ([46],[47]) and to reduce waste leading to lean organization [48]. These organizations heavily use Information and Communication Technologies (ICT) leading to increase the call for IT inter-operability. This has lead to several EU FP6 interoperability projects (e.g. INTEROP NoE, ATHENA IP) and to the Enterprise Interoperability Roadmap [49]. These works have mostly focused on enterprise engineering methods (e.g. [50] provides a federative architecture to support engineering, and [51] focuses on process models) and ICT implementation constraints (see the ATHENA Service Oriented Interoperability reference architecture [52]) and are rather similar to Enterprise Architecture Frameworks (e.g. TOGAF, DoDAF, FEAF) designed to support Information system interoperability thanks to services and design methodologies based on reusable components. Only few works ([53], [54], [55]) have paid attention to actors interactions leading to design global evolution models specifically for SME collaborative networks, leading to adapt existing enterprise engineering frameworks ([56]).

Over the past decades researchers have directed their attention toward services oriented business models in which intangible resources, exchange processes, and relationships are central ([57],[35]). Traditional economic models/methods for describing operational and financial activities of a firm as part of a complex service system, are inadequate in the emergence of a services world in which the focus is now on the customer rather than the producer. In this context, as cooperation/interaction among different business roles in the process of providing a service is a key element in understanding and observing service systems, the field of game theory becomes a useful tool for identifying rules and strategies that optimize business objectives. Game theory has been applied in different environments such as social networks ([58],[59]), Internet-like networks ([60]), telecommunication networks ([61],[62]) and transportation networks ([63],[64]). In [59], a dynamic social network model is considered, in which agents play repeated games in pairings determined by a stochastically evolving social network. Basic steps are taken in exploring dynamics of evolution of interaction structures and co-evolution of structure and strategy. The ultimate goals are to create models that are more true to life, and to find theoretical bases for observed behaviors of systems, including prediction of selection between multiple equilibria. In [60], a game-theoretic model of network creation is proposed. Selfish node-agents pay for the links that they establish and benefit in quality of service improvements. The network nodes comprise one type of players rather than forming a service system of different entities that complement each other in the process of providing a service. In [61], a network of Internet service providers, intermediaries and end users is defined and analyzed. A hierarchically structured model is proposed, that uses synchronized auction mechanisms in order to allocate bandwidth efficiently to the end users. This study is a simplified case of a service system and an attempt to model players' actions on the basis of game theoretic mechanisms is undertaken. In [63], an agent-based model is developed to study the welfare consequences of alternative ownership policies for road networks. The evolution of a transportation network consisting of profit-maximizing private roads in a market economy is analyzed and modeled. One limitation of this study is that cooperation among private roads is not taken into consideration. The questions of when and how a link seeks coalitions at the microscopic level and how road network evolves with changing ownership structures need to be answered.

There are a few studies that use game theoretic approaches in complex service systems ([65], [66]). In [65], a framework for the modeling and analysis of business model is proposed involving a network of interconnected business entities. The framework includes an ecosystem-modeling component, a simulation component, and a service analysis component, and integrates methods from value network modeling, game theory analysis, and multi-agent systems. At the analysis level, more work is required to develop a holistic approach that can easily allow for different perspectives and different life-cycle stages of the model elements, as well as allow for a larger number of decision-interaction combinations.

5.3 Simulations of ASNs

There is a need to investigate new multilevel concepts for tools for a data-intensive science of large scale systems with ICT tightly entangled with business structures. There are few works that develop systematic means to gain knowledge of the service sector systems and to model, predict and characterise their behaviour, their dynamics and their evolution ([39], [43], [44]).

In [39], an approach to analyze and reconfigure value networks in a qualitatively manner is proposed that takes into account the role of knowledge and intangible value exchange as the foundation for these emerging networked enterprises. In this approach, mapping a value network involves diagramming all value exchanges with each member of the business or organizational network. This methodology is applied to a pharmaceutical company. The analysis revealed that even though the company respected its financial relationship with medical providers, it neglected knowledge exchanges, which were handled inconsistently across it.

In [43], the e3-value methodology is applied to a news service called "Amsterdam Times" which offers to all its subscribers the ability to read articles online. The economic feasibility of this idea is evaluated in quantitative terms that are based on an assessment of the value of objects for all entities involved. The article focuses mainly on building confidence that an e-business idea is of real interest for all actors involved. Additionaly, it analyzes what-if scenarios, which can help companies understand the sensitivity of e-business models with respect to financial parameters, future trends, and other parameters such as customer behaviour.

In [44], the proposed approach is applied in a repair service system as part of the automotive industry. Dealers, manufacturers, and their suppliers collaborate in order to satisfy customer requests. The manufacturer generates parts catalogues that are delivered to the dealers and suppliers every month. Business models are defined and the value created by the various partners is computed. In order to increase the value of the above system, a transformation of the traditional service system is proposed. In the new system a central portal is created by the manufacturer or an outsourcer that provides up-to-date information (the content of catalogues) that can be accessed by any partner. Under these conditions, repair time is reduced, thus customer satisfaction is increased leading to more sales. Additionally, it is shown that the costs of creating the information system or paying an outsourcer for providing it, are less than the catalogue generation and delivery costs. Thus, the total value is increased.

6 Summary

This report presented the state of the art in Business Process Management for SOA-enabled business processes. It reviewed current BPM technologies and practices, highlighted their main characteristics and limitations and introduced the concept of Agile Service Networks as a fundamental step in the transition to the next-generation of service-enabled BPM technology.

The report first introduced the concepts of business process, process automation, and workflows. Virtually all enterprises have business processes that require process automation. As such, any process automation tool should be able to easily control and coordinate activities and provide an easy method to define the business process and the underlying flows of information between applications. Process automation is different from traditional document workflow in the sense that it involves integration between computer-based systems and manual steps and tasks. It is implemented for automating information streams between applications to enable business processes. Traditional workflow tools focus instead on handling the movement of documents between people who are required to perform tasks on these documents. These processes comprise both automated and manual tasks.

Traditional Business Process Management combines process automation together with task-based workflow into a managed, end-to-end process within the confines of a single organization. BPM codifies value-driven processes and institutionalizes their execution within the enterprise. This implies that BPM tools can help analyze, define and enforce process standardization. BPM thrives on modeling tools to visually construct, analyze and execute cross-functional business processes within the enterprise. Design and modeling of business processes is accomplished by means of sophisticated graphical tools. Combining BPM with real-time analysis of logs and events generated by the processes allows business analysts to track and understand how business processes execute at any given point in time.

The interplay of BPM technologies with Service Oriented Architectures (SOAs) provides a flexible infrastructure that enables exposing independent reusable business processes as services, which can be dynamically enacted and combined according to the business needs. To this end, the report placed special emphasis on business protocols, reusability of process fragments and transactional business processes.

The report concluded that the current generation BPM technologies do not address adequately the requirements of complex applications involving several organizations that need to collaborate in a dynamic manner. Current BPM deployments are narrow in scope, adopt an organization-centric view providing only improvements in specific business functions within the confines of a single organization, and as a result BPM suites only enable organizations to enhance their existing processes.

One issue which received particular interest was the use of advanced technologies and higher-level abstraction mechanisms in order to create flexible end-to-end process constellations that span organizational boundaries and are referred to as Agile Service Networks. Service networks comprise large numbers of long-running, highly dynamic complex end-to-end service interactions reflecting asynchronous message flows that typically transcend several organizations and span several geographical locations. ASNs essentially provide much more functionality and flexibility when compared to traditional BPM, enabling organizations to innovate new value delivery systems that transcend the enterprise and extend to every external partner. They help achieve the transition from a relatively static view of an organization to a much more dynamic, high-value one where end-to-end business process interactions and trends are examined much more closely to understand much more accurately the business dynamics. Using ASNs, organizations can deploy, monitor, and continuously update cross-enterprise functions within a mixed environment of people, content, and systems.

7 References

[1] A. U. Mallya, and M. P. Singh, "A semantic approach for designing e-business protocols," *ICWS '04: Proceedings of the IEEE International Conference on Web Services*, p. 742, 2004.

[2] S. Barat, V. Kulkarni, and D. Janakiram, "A safety criterion for reusing a business process in the desired integrated," *SCC '06: Proceedings of the IEEE International Conference on Services Computing*, pp. 381—389, 2006.

[3] B. Benatallah, F. Casati, and F. Toumani, "Web service conversation modeling: A cornerstone for e-business automation," *IEEE Internet Computing*, vol. 8, no. 1, pp. 46—54, January/February, 2004.

[4] T. Bultan, X. Fu, R. Hull, and Jianwen Su, "Conversation specification: A new approach to design and analysis of e-service composition," *WWW '03: Proceedings of the 12th International Conference on World Wide Web*, pp. 403—410, 2004.

[5] B. Benatallah, F. Casati, and F. Toumani, "Representing, analysing and managing web service protocols," *Data and Knowledge Engineering Journal*, vol. 58, no. 3, pp. 327—357, 2006.

[6] L. Bordeaux, G. Salaün, D. Berardi, and M. Mecella, "When are two web services compatible?," *TES '04: 5th International Workshop on Technologies for E-Services*, pp. 15—28, 2005.

[7] B. Benatallah, F. Casati, F. Toumani, J. Ponge, H. Reza, and M. Nezhad, "Service Mosaic: A model-driven framework for web services life-cycle management," *IEEE Internet Computing*, vol. 10, no. 4, pp. 55—63, 2006.

[8] D. Beyer, A. Chakrabarti, and T. A. Henzinger, "Web service interfaces," *WWW '05: Proceedings of the 14th International Conference on World Wide Web*, pp. 148—159, 2005.

[9] J. Ponge, B. Benatallah, F. Casati, and F. Toumani, "Fine-grained compatibility and replaceability analysis of timed web service protocols," *ER '07: Proceedings of the International Conference on Conceptual Modeling*, pp. 599—614, 2007.

[10] N. Desai, and M. P. Singh, "Protocol-based business process modeling and enactment," *ICWS '04: Proceedings of the IEEE International Conference on Web Services (ICWS'04)*, pp. 35—42, 2004.

[11] N. Desai, A. U. Mallya, A. K. Chopra, and M. P. Singh, "Interaction protocols as design abstractions for business processes," *IEEE Transactions on Software Engineering*, vol. 31, no. 12, pp. 1015—1027, December 2005.

[12] N. Desai, A. K. Chopra, and M. P. Singh, "Business process adaptations via protocols," *SCC '06: Proceedings of the IEEE International Conference on Services Computing*, pp. 103—110, 2006.

[13] A. Svirskas, J. Bedzinskas, B. Roberts, I. Ignatiadis, and M. Wilson, "Adaptive business protocol support in service oriented B2B collaborations," *WETICE '06: Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 261—262, 2006.

[14] M. Oya, M. Kinoshita, and Y. Kakazu, "On dynamic generation of business protocol in autonomous web services," *Systems and Computers in Japan*, vol. 37, no. 2, pp. 37—45, 2006.

[15] N. Guermouche, S. Benbernou, E. Coquery, and M. S. Hacid, "Privacy-aware web service protocol replaceability," *ICWS '07: Proceedings of the IEEE International Conference on Web Services*, pp. 1048—1055, 2007.

[16] C. Li, J. Gou, H. Wu, and G. Li, "A process meta-model supporting domain reuse," *2005 International Software Process Workshop*, pp. 459—461, 2005.

[17] L. Liao, and H. K. N. Leung. "An ontology-based business process modeling methodology," *ACST '07: Proceedings of the 3rd conference on IASTED International Conference*, pp. 114—119, 2007.

[18] I. Markovic, and A. Costa Pereira, "Towards a formal framework for reuse in business process modeling," *BPM '07 International Workshops*, pp. 484—495, 2007.

[19] K. Mokhtari, S. Benbernou, M. Said, E. Coquery, M. S. Hacid, and F. Leymann, "Verification of privacy timed properties in web service protocols," *SSC '08: Proceedings of the 7th International ITG Conference on Source and Channel Coding*, to appear.

[20] S. H. Ryu, R. Saint-Paul, B. Benatallah, and F. Casati, "A framework for managing the evolution of business protocols in web services," *APCCM '07: Proceedings of the 4th Asia-Pacific Conference on Conceptual Modeling*, pp. 49—59, 2007.

[21] Y. Mou, J. Cao, and S. Zhang, "A process component model for enterprise business knowledge reuse," *SCC '04: Proceedings of the IEEE International Conference on Services Computing*, pp. 409—412, 2004.

[22] N. Russell, A. H. M. ter Hofstede, W. M. P. van der Aalst, and N. Mulyar, "Workflow control-flow patterns: A revised view," *BPM Center Report BPM-06-22*, 2006.

[23] J. Eder, and W. Liebhart, "Workflow Recovery," Proceedings of the First IFCIS International Conference on Cooperative Information Systems, pp. 124—134, 1996.

[24] S. Jablonski, and C. Bußler, *Workflow management: modeling concepts, architecture and implementation*. Bonn: International Thomson Publishing, 1996.

[25] F. Leymann, and D. Roller, *Production workflow: Concepts and techniques*. Upper Saddle River: Prentice Hall, 2000.

[26] W. M. P. van der Aalst, and M. Pesic, "DecSerFlow: Towards a truly declarative service flow language," *WF-FM '06: Proceedings of the 3rd International Workshop on Web Services and Formal Methods*, pp. 1—23, 2006.

[27] W. M. P. van der Aalst, M. Weske, and D. Grünbauer, "Case handling: A new paradigm for business process support," *Data and Knowledge Engineering Journal*, vol. 53, no. 2, pp. 129—162, 2005.

[28] M. Adams, A. H. M. ter Hofstede, D. Edmond, and W. M. P. van der Aalst, "Worklets: A service-oriented implementation of dynamic flexibility in workflows," *CoopIS 2006: On the Move to Meaningful Internet Systems 2006, OTM Confederated International Conferences, 14th International Conference on Cooperative Information Systems*, pages 291—308, 2006.

[29] M. Reichert, and P. Dadam, "Adept$_{flex}$: Supporting dynamic changes of workflows without losing control," *Journal of Intelligent Information* Systems, vol. 10, no. 2, March/April 1998.

[30] D. Karastoyanova, and A. Buchmann, "ReFFlow: A model and generic approach to flexibility of web service compositions," *iiWAS '04: Proceedings of the International Conference on Information Integration and Web-based Applications and Service*, 2004.

[31] N. Russell, A.H.M. ter Hofstede, D. Edmond, and W. M. P. van der Aalst, "Workflow data patterns," Queensland University of Technology, Brisbane, QUT Technical report, FIT-TR-2004-01, 2004.

[32] N. Russell, A. H. M. ter Hofstede, D. Edmond, and W. M. P. van der Aalst, "Workflow resource patterns," Eindhoven University of Technology, Eindhoven, BETA Working Paper Series, WP 127, 2004.

[33] N. Russell, W. M. P. van der Aalst, and A. H. M. ter Hofstede, "Exception handling patterns in process-aware information systems," BPMcenter.org, BPM Center Report BPM-06-04, 2006.

[34] S. E. Sampson, Understanding service businesses: Applying principles of the unified services theory, 2nd ed. New York: John Wiley & Sons, 2001.

[35] J. Spohrer, P. Maglio, J. Bailey, and D. Gruhl, "Steps towards a science of service systems," *IEEE Internet Computing*, vol. 40, no. 1, pp. 71—77, 2007.

[36] M. Iansiti, and R. Levien, The keystone advantage: What the new dynamics of business ecosystems mean for strategy, innovation and sustainability. Harvard University School Press, 2004.

[37] C. Parolini, The value net: A tool for competitive strategy. John Wiley & Sons, 1999.

[38] V. Allee, The future of knowledge: Increasing prosperity through value networks. Boston: Butterworth-Heinemann, 2002.

[39] V. Allee, "Reconfiguring the value network," *Journal of Business Strategy*, vol. 21, no. 4, July/August 2000.

[40] Y. M. Lee, F. Cheng, and Y. T. Leung, "Exploring the impact of RFID on supply chain dynamics," *WSC '04: Proceedings of the 36th Conference on Winter Simulation*, pp. 1145—1152, 2004.

[41] A. Verna, "Understanding Value Networks". 2002. [Online] Available: www.vernaallee.com/value_networks/Understanding_Value_Networks.html [Accessed June 25, 2008].

[42] R. Normann, and R. Ramirez, *Designing Interactive Strategy from Value Chain to value constellation*. Wiley, 2005.

[43] J. Gordijn, and H. Akkermans, "Designing and evaluating e-business models," *IEEE Intelligent Systems*, vol. 16, no. 4, pp. 11—17, 2001.

[44] N. Caswell, et al., "Estimating value in value networks," *IBM Systems Journal*, vol. 47, no. 1, 2008.

[45] M. O. Jackson, and F. Bloch, "The formation of networks with transfers among players," FEEM Working Paper no. 80.04, May 2004.S. Goldman, R. Nagel, and K. Preiss, *Agile Competitors and Virtual Organizations*. New York: Van Nostrand Reinhold, 1995.

[46] S. Goldman, R. Nagel, and K. Preiss, *Agile Competitors and Virtual Organizations*. New York: Van Nostrand Reinhold, 1995.

[47] H. L. Lee, "The Triple-A supply chain," *Harvard Business Review*, pp. 102—112, October 2004.

[48] J. P. Womack, and D. T. Jones, *Lean Thinking*, 2nd ed. Simon & Schuster, 2003.

[49] M. S. Li, R. Cabral, G. Doumeingts, and K. Poplewell, *Enterprise Interoperability Roadmap*, 2006. [Online]. Available: ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate_d/ebusiness/ei-roadmap-final_en.pdf [Accessed June 25, 2008].

[50] "GERAM: Generalized Enterprise Reference Architecture and Methodology", IFAC-IFIP, v. 1.6.3, IFAC-IFIP Task Force on Architecture and Methodology, March 1999.

[51] A. W. Sheer, "Architecture of Integrated Information System (ARIS)", DIISM '93: Proceedings of the JSPE/IFIP TC5/WG5.3 Workshop on the Design of Information Infrastructure Systems for Manufacturing, pp. 85—99, 1993.

[52] The ATHENA Consortium, "Interoperability reference architecture," March 2007. [Online]. Available: http://modelbased.net/aif/reference_architecture/reference_model.html [Accessed June 25, 2008].

[53] H. W. Kim, "Business process versus coordination process in organizational change", *The international journal of flexible manufacturing systems*, vol. 12, pp. 275—290, 2000.

[54] C. Lackenby, and H. Seddighi, "A dynamic model of virtual organizations: formation and development," *Collaborative Business Ecosystems and Virtual Enterprises*, pp. 37—44, 2000.

[55] P. Burlat, D. Vila, B. Besombes, and V. Deslandres, "Un cadre de modélisation des trajectoires d'évolution des groupements d'entreprises," *Actes de CIGI'01*, June 2001.

[56] F. Biennier, X. Boucher, A. Hammami, and L. Vincent, "Towards a modeling framework for networks of SMEs," *Collaborative Business Ecosystems and Virtual Enterprises*, pp. 11—18, 2002.

[57] S. Largo, and R. Lusch, "Evolving to a new dominant logic for marketing," *Journal of Marketing*, vol. 68, no. 1, pp. 1—17, 2004.

[58] B. Skyrms, and R. Pemantle, “A dynamic model of social network formation,” *Proceedings of the National Academy of Sciences USA*, 9340—9346, 1997.

[59] B. Buskens, *Social Networks and Trust*. Springer, 2002.

[60] A. Fabrikant, A. Luthra, E. Maneva, C. Papadimitriou, and S. Shenker, “On a network creation game,” *Proceedings of the 22nd annual symposium on Principles of distributed computing*, 347—351, 2003.

[61] M. Bitsaki, G. D. Stamoulis, and C. Courcoubetis, "An efficient auction-based mechanism for hierarchically structured bandwidth markets," *Computer Communications*, vol. 29, no. 7, pp. 911—921, 2006.

[62] G. Camponovo, and Y. Pigneur, "Analyzing the actor game in m-business," *Proceedings of the 1st International Conference on Mobile Business*, 2002.

[63] L. Zhang, and D. Levinson, “Economics of road network ownership,” University of Minnesota: Nexus Research Group, Working Paper 000009, 2006.

[64] N. Montes de Oca, F. Xie, and D. Levinson, "Forecasting and evaluating network growth," University of Minnesota: Nexus Research Group, Working Paper 000010, 2006.

[65] C. Tian, B. Ray, J. Lee, R. Cao, and W. Ding, “BEAM: A framework for business ecosystem analysis and modeling,” *IBM Systems Journal*, vol. 47, no. 1, February 2008.

[66] R. Kauffma, and A. Kumar, “Modeling network decisions under uncertainty: Countervailing externalities and embedded options,” *HICSS '07: Proceedings of the 40th Hawaii International Conference on System Sciences*, p. 209c, 2007.

[67] A. Neely, M Gregory, and K. Platts, “Performance measurement system design: A literature review and research agenda,” *International Journal of Operations & Production Management*, vol. 25, no. 12, pp. 1228—1263, 2005.

[68] *SLA Management Handbook*, vol. 4, The Open Group, Enterprise Perspective, 2004.

[69] V. Popova, and J. Treur, “A Specification language for organizational performance indicators,” *AIAI '07: Proceedings of the 18th International Conference on Innovations in Applied Artificial Intelligence*, pp. 291—301, 2007.

[70] R. Kaplan, and D. Norton, “The balanced scorecard: Measures that drive performance,” *Harvard Business Review*, pp. 71—79, January/February 1992.

[71] A. Gunasekarana, C. Patelb, and R. E. McGaugheyc, “A framework for supply chain performance measurement,” *International Journal of Production Economics*, vol. 87, pp. 333—347, 2004.

[72] P. C. Brewer, T. W. Speh, “Using the balanced scorecard to measure supply chain performance,” *Journal of Business Logistics*, vol. 21, no. 1, pp. 75—93, 2000.

[73] J. P. C. Kleijnen, and M. T. Smits, “Performance metrics in supply chain management,” *Journal of the Operational Research Society*, vol. 54, no. 5, pp. 507—514, 2003.

[74] M. zur Muehlen, and M. Rosemann, “Workflow-based process monitoring and controlling: Technical and organizational issues,” *HICSS '00: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, p. 10, 2000.

[75] M. Castellanos, F. Casati, M. Shan, and U. Dayal, “iBOM: A platform for intelligent business operation management,” *ICDE '05: Proceedings of the 21st International Conference on Data Engineering*, pp. 1084—1095, 2005.

[76] P. Chowdhary, et al., “Model driven development for business performance management,” *IBM System Journal*, vol. 45, no. 3, 2006.

[77] C. Momm, R. Malec, S. Abeck, “Towards a model-driven development of monitored processes,” *Wirtschaftsinformatik*, no. 2, pp. 319—336, 2007.

[78] H. Roth, J. Schiefer, A. Schatten, “Probing and monitoring of WSBPEL processes with web services,” CEC-EEE: Proceedings of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, p. 30, 2006.

[79] L. Baresi, S. Guinea, “Towards dynamic monitoring of WS-BPEL processes,” *ICSOC '05: Proceedings of the 3rd International Conference on Service-Oriented Computing*, pp. 269—282, 2005.

[80] T. Andrews et. al. (eds), "Business Process Execution Language for Web Services," May 2003. [Online]. Available: IBM, http://www.ibm.com/developerworks/library/ws-bpel

[81] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D. F. Ferguson, "SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More" Prentice Hall, 2005.

[82] M. Kloppmann, D. Koenig, F. Leymann, G. Pfau, A. Rickayzen, C. von Riegen, P. Schmidt, and I. Trickovic, "WS-BPEL extension for people: BPEL4People," IBM and SAP, joint white paper, July 2005. [Online]. Available: IBM, http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/BPEL4People_v1.pdf

[83] D. Chappell, "Enterprise Service Bus", O'Reilly, 2004.

[84] Y. Han, A. Sheth, and C. Bussler, "A taxonomy of adaptive workflow management," *Workshop of the 1998 ACM Conference on Computer Supported Cooperative Work*, 1998.

[85] W. M. P. van der Aalst, and S. Jablonski, "Dealing with workflow change: identification of issues and solutions," *International Journal of Computer Systems Science and Engineering*, vol. 15, no. 5, pp. 267—276, September 2000.

[86] M. Reichert, and P. Dadam, "A framework for dynamic changes in workflow management systems," *DEXA '97: Proceedings of 8th International Workshop on Database and Expert Systems Applications*, p. 42, 1997.

[87] D. Karastoyanova, A. Houspanossian, M. Cilia, F. Leymann, and A. Buchmann, "Extending BPEL for run time adaptability," *EDOC '05: Proceedings of the 9th IEEE International EDOC Enterprise Computing Conference*, pp. 15—26, 2005.

[88] D. Karastoyanova, F. Leymann, J. Nitzsche, B. Wetzstein, and D. Wutke, "Parameterized BPEL processes: Concepts and implementation," *BPM '06: Proceedings of the 4th International Conference on Business Process Management*, pp. 471—476, 2006.

[89] J. Nitzsche, T. van Lessen, D. Karastoyanova, and F. Leymann, "BPEL for semantic web services," *AweSome '07: Proceedings of the 3rd International Workshop on Agents and Web Services in Distributed Environments*, pp. 179—188, 2007.

[90] T. van Lessen, J. Nitzsche, M. Dimitrov, M. Konstantinov, D. Karastoyanova, and L. Cekov, "An execution engine for semantic business process," *SeMSoC '07: 2nd International Workshop on Business Oriented Aspects Concerning Semantics and Methodologies in Service-oriented Computing*, 2007.

[91] M. P. Papazoglou, and P. M. A. Ribbers, "e-Business: Organizational and Technical Foundations". J. Wiley & Sons, April 2006.

[92] T. H. Davenport, "The e-process evolution," *Business Process Management Journal*, vol. 10, no. 1, 2004.

[93] P. Harmon "Analyzing activities," *Business Process Trends*, vol. 1, no. 4, April 2003.

[94] B. von Halle, *Business Rules Applied*, J. Wiley & Sons, 2001.

[95] R. G. Ross, *Principles of the Business Rule Approach*, Addison-Wesley Information Technology Series. Addison-Wesley, 2003.

[96] D. McGoveran, "An Introduction to BPM and BPMS," *Business Integration Journal*, pp. 2—10, April 2004.

[97] E. Roch "Application integration: Business and technology trends," *EAI Journal*, August 2002.

[98] F. H. Nah, M. Rosemann, and E. Watson, "E-Business Process Management," *Business Process Management Journal*, vol. 10, no. 1, 2004.

[99] A. Charfi, R. Khalaf, and N. Mukhi, "QoS-aware web service compositions using non-intrusive policy attachment to BPEL," *ICSOC '07: Proceedings of the International Conference on Service Oriented Computing*, pp. 582—593, 2007.

[100] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D. F. Ferguson: *Web Services Platform Architecture*. Prentice Hall, 2005.

[101] M. P. Papazoglou, *Web services: Principles and technology*. Prentice-Hall, 2007.

[102] M. P. Papazoglou, and F. Leymann, "Business process management," *IEEE Encyclopedia of Software Engineering*, 2008.

[103] M. P. Papazoglou, "Web services and business transactions," *World Wide Web*, vol. 6, no. 1, pp. 49--91, 2003.

[104] F. Tartanoglu, V. Issarny, A. Romanovsky, and N. Levy, "Coordinated forward error recovery for composite web services," *Proceedings of the 22nd IEEE Symposium on Reliable Distributed Systems*, 2003.

[105] J. Cardoso, A. Sheth, J. Miller, J. Arnold, and K. Kochut, "Quality of service for workflows and web service processes," *Journal of Web Semantics*, vol. 1, no. 3, pp. 166—206, 2004.

[106] S. Tai, R. Khalaf, and T. Mikalsen, "Composition of coordinated web services," *Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*, pp. 294—310, 2004.

[107] M. Brambilla, G. Guglielmetti, and C. Tziviskou, "Asynchronous web services communication patterns in business protocols," *WiSE '05: Proceedings of the 6th International Conference on Web Information Systems Engineering*, pp. 435—442, 2005.

[108] M. Paludo, R. Burnett, and E. Jamhour. "Patterns leveraging analysis reuse business processes," *ICSR-6: Proceedings of the 6th International Conference on Software Reuse: Advances in Software Reusability*, pp. 353—368, 200.

[109] Z. Ma, and F. Leymann, "A lifecycle model for using process fragment in business process modeling," *BPMDS '08: Proceeding of the 9th Workshop on Business Process Modeling, Development, and Support*, to appear.

[110] C. Ouyang, M. Dumas, S. Breutel, and A. ter Hofstede, "Translating standard process models to BPEL," *Advanced Information Systems Engineering*, vol. 4001, pp. 417—432, 2006.

[111] W. Zhao, R. Hauser, K. Bhattacharya, B. R. Bryant, and F. Cao, "Compiling business processes: untangling unstructured loops in irreducible flow graphs," *International Journal of Web and Grid Services*, vol. 2, pp. 68—91, 2006.

[112] J. Koehler, R. Hauser, J. Küster, K. Ryndina, J. Vanhatalo, and M. Wahler, "The role of visual modeling and model transformations in business-driven development," *GT-VMT 2006: Proceedings of the Fifth International Workshop on Graph Transformation and Visual Modeling Techniques*, 2006.

[113] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-oriented computing: State of the art and research challenges", *IEEE Internet Computing*, vol. 40, no. 11, pp. 38—45, 2007.

[114] A. Arsanjani, "Service-oriented modeling and architecture," *IBM developerWorks*, November 2004.

[115] A. Charfi, M. Mezini, "Hybrid web service composition: business processes meet business rules", in *Proceedings of the 2nd international conference on Service oriented computing*, 2004, pp. 30-38.

[116] T. Debevoise, "Business Process Management With a Business Rules Approach: Implementing the Service Oriented Architecture", Business Knowledge Architects, 2005.

[117] F. Rosenberg, S. Dustdar, "Business Rules Integration in BPEL - A Service-Oriented Approach", In *CEC '05: Proceedings of the Seventh IEEE International Conference on E-Commerce Technology*, 2005, pp. 476–479.

[118] B. Orriens, J. Yang, M.P. Papazoglou, "A rule driven approach for developing adaptive service oriented business collaboration", in *Proceedings of the 3rd international conference on Service oriented computing*, December 2005, pp. 182-189.

[119] S. Goedertier and J. Vanthienen, "Compliant and Flexible Business Processes with Business Rules", *7th Workshop on Business Process Modeling, Development and Support (BPMDS'06) at CAiSE'06*, Presses Universitaires de Namur, 2006, pp. 94 - 104.

[120] S. Sadiq, G. Governatori, K. Namiri, "Modelling Control Objectives for Business Process Compliance", *5th International Conference on Business Process Management*, 24-28 September 2007, Brisbane Australia.

[121] R. Lu, S. Sadiq, G. Governatori, "Compliance Aware Business Process Design", *3rd International Workshop on Business Process Design (BPD'07). In conjunction with the 5th International Conference on Business Process Management*, 24-28 September 2007, Brisbane Australia.

[122] C. Giblin, S. Muller, B. Pfitzmann, "From regulatory policies to event monitoring rules: Towards model driven compliance automation", IBM Research Report, Zurich Research Laboratory, October 2006.

[123] S. Goedertier, J. Vanthienen, "Designing Compliant Business Processes with Obligations and Permissions", *J. Eder, S. Dustdar et al. (Eds.) BPM 2006 Workshops, LNCS 4103*, Springer Verlag, 2006.

[124] M. zur Muehlen, M. Rosemann, "Integrating Risks in Business Process Models", *16th Australasian Conference on Information Systems*, Sydney, Australia, 29 November – 2 December 2005.

[125] M.P. Papazoglou, "The challenges of service evolution", In: *Proceedings of Advanced Information Systems Engineering Conference: CAISE 2008*, Springer-Verlag, keynote address, 2008, pp. 1-15.

[126] M.P. Papazoglou, "Compliance requirements for business-process driven SOAs", In: *Proceedings of IFIP World Computer Congress Conference: WCC 2008*, Springer-Verlag, keynote address, 2008.

[127] D. Webber, "Understanding ebxml, uddi and xml/edi" http://www.touchbriefings.com/pdf/967/59.pdf, 2000.

[128] B. Meadows, L. Seaburg, "Universal Business Language 1.0" http://www.docs.oasis-open.org/ubl/cd-UBL-1.0/, 2004.

[129] P. Hrastnik, W. Winiwarter, "Using advanced transaction meta-models for creating transaction-aware web service environments" *International Journal of Web Information Systems 1(2)*, 2005, pp. 89–99.

[130] H. Ludwig, H. Gimpel, A. Dan, B. Kearney, "Template based automated service provisioning supporting the agreement driven service life-cycle", In: *B. Benatallah,F. Casati, P. Traverso (eds) Service-oriented computing - ICSOC 2005, 3rd Proceeding of international conference, Amsterdam, Lecture Notes in Computer Science, vol. 3826*, Springer, Berlin, 2005, pp. 283–295.

[131] A. Andrieux, K, Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, M. Xu, "Web services agreement specification (ws-agreement)", Technical report, Grid Resource Allocation Agreement Protocol (GRAAP) WG, 2005.

[132] J. O'Sullivan, D. Edmond, A.H.M. ter Hofstede, "Formal description of non-functional service descriptions", Technical report, Queensland University of Technology http://www.bpm.fit.qut.edu.au/about/docs/non-functional.jsp, 2005.

[133] T. Wang, J. Vonk, P. Grefen, "TxQoS: A Contractual Approach for Transaction Management", *11th IEEE International1Enterprise Distributed Object Computing Conference*, 2007, pp. 327 – 327.

[134] M.P. Papazoglou and B. Kratz, "Web services technology in support of business transactions", *Service Oriented Computing and Applications, 1(1)*, Springer-Verlag Berlin Heidelberg, 2007, pp. 51-63.

[135] M. Weske, "Flexible Modeling and Execution of Workflow Activities", submitted to *31st Hawaii International Conference on System Sciences (HICSS-31)*, 1998.

[136] W. Deiters, V. Gruhn, "The FUNSOFT Net Approach to Software Process Management". *International Journal on Software Engineering and Knowledge Engineering, Vol. 4, No. 2*, 1994.

[137] M. Hsu, C. Kleissner, "ObjectFlow: Towards a Process Management Infrastructure", *Distributed and Parallel Databases, Vol. 4*, 1996.

[138] A. Charfi, M. Mezini, "Aspect-Oriented Web Service Composition with AO4BPEL", In *Proceedings of ECOWS 2004*.

[139] J. Mendling, M. Nttgens, "Exchanging EPC business process models with EPML," in *Proceedings of the 1st GI Workshop XML Interchange Formats for Business Process Management*, March 2004.

[140] G. Booch, J. Rumbaugh, and I. Jacobson, "The Unified Modeling Language User Guide", Addison-Wesley, 1998.

[141] Object Management Group, "Business process modeling notation," February 2006. [Online]. Available: http://www.bpmn.org/Documents/OMG%20Final%20Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf

[142] ebXML Initiative, "ebXML business process specification schema version 1.01," May 2001. [Online]. Available: http://www.ebxml.org/specs/ebBPSS.pdf

[143] J. Dubray, "A novel approach for modeling business process definitions," 2003. [Online]. Available: http://www.ebpml.org

[144] N. Jennings, P. Faratin, T. Norman, P. O'Brien and B. Odgers, "Autonomous Agents for Business Process Management," *International Journal of Applied Artificial Intelligence, vol. 14, no. 2*, pp. 145–189, 1996.

[145] DAML-S Coalition, "Daml-s," May 2003. [Online]. Available: http://www.daml.org/services/daml-s/0.9/

[146] W. van der Aalst, "The application of petri nets to workflow management," *Journal of Circuits, Systems and Computers, vol. 8, no. 1*, pp. 21–66, 1998.

[147] R. Hamadi and B. Benatallah, "A petri net-based model for web service composition," in Proceedings of the 14th Australasian Database Conference, February 2003.

[148] J. Bergstra, A. Ponse, and S. Smolka, 'Handbook of Process Algebra", Elsevier, 2001.

[149] R. Milner, "Operational and Algebraic Semantics of Concurrent Processes", Elsevier, 1993.

[150] C. Hoare, "Communicating Sequential Processes", Prentice Hall, 1985.

[151] J. Bergstra, J. Klop, "Algebra of communicating processes with abstraction," Theoretical Computer Science, vol. 37, no. 1, pp. 77–121, 1985.

[152] F. Arbab, "Reo: a Channel-based Coordination Model for Component Composition," Mathematical Structures in Computer Science, vol. 14, no. 3, 2004.

[153] OASIS WS-BPEL TC, "Web Services Business Process Execution Language, Version 2.0", 2007. http://docs.oasis-open.org/wsbpel/2.0/CS01/wsbpel-v2.0-CS01.html

[154] C.W. Krueger, "Software reuse", *ACM Computing Surveys, vol. 24 (2)*, 1992, pp. 131 – 183.

[155] A. Charfi, B. Schmeling, and M. Mezini, "Transactional bpel processes with ao4bpel aspects," in Proceedings of the Fifth European Conference on Web Services, 2007.

[156] M. Papazoglou and B. Kratz, "A business-aware web services transactions model," in Proceedings of the 4th International Conference on Service-Oriented Computing, A. Dan and W. Lamersdorf, Eds. Springer-Verlag, 2006, pp. 352–364.