Tijdschrift voor Economie en Management
Vol. L, 4, 2005

Business Process Management: a Bird's-Eye View and Research Agenda

by M. SNOECK and W. LEMAHIEU

Monique Snoeck
KULeuven, Department of Applied Economics, Leuven

Wilfried Lemahieu
KULeuven, Department of Applied Economics, Leuven

ABSTRACT

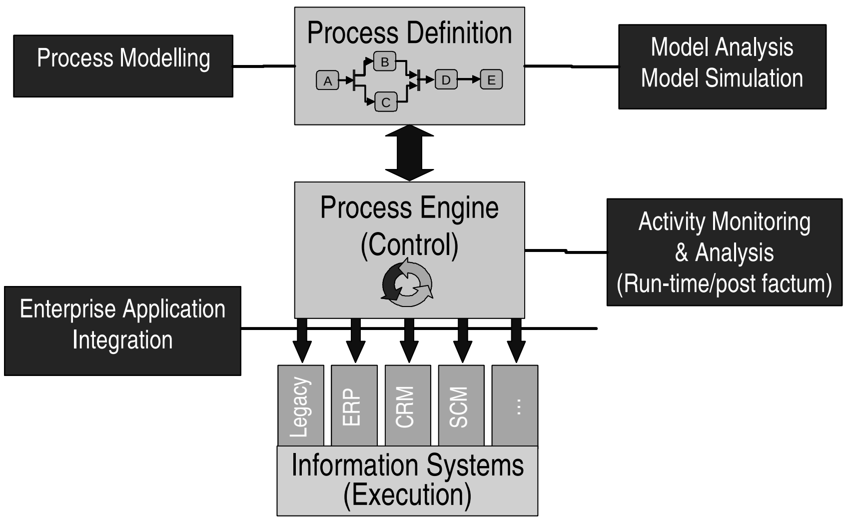
Recently, Business Process Management (BPM) has gained a lot of attention both from the academic world and the industrial world. The similarities between workflow management and business process management can make one wonder whether it's all old wine in new bottles. But although there are many similarities and many results from workflow research still apply, recent technological developments have changed the domain in a substantial way, resulting in new opportunities and interesting research questions. The convergence of workflow management and Enterprise Application Integration yields prospects for integrating the organisational view and the engineering or automation view of an information system. Internet technologies such as XML and SOAP offer the potential of massive scalability and interconnectivity. In this paper, we first provide a broad overview of the domain of BPM and identify challenges from a modelling quality perspective. Thereafter, we zoom into the use of web services technology for business processes enactment. Furthermore, we review some obvious and less obvious application domains of business process management and business process enactment. Finally, we suggest a research agenda for BPM, by proposing some open issues.

I. WHAT IS BUSINESS PROCESS MANAGEMENT ?

A business process is a sequence of steps executed in order to realise a business goal. Depending on the domain one speaks of customer processes, supplier processes, human resource processes, financial processes, procurement processes, e-government processes … and so on. In other words, one can say that business processes constitute the essence of the administrative organisation of an enterprise.

The term Business Process Management refers to developments in the areas of modelling, analysis, improvement, and enactment of business processes. Figure 1 gives an overview of the different sub-domains in the area of Business Process Management.

FIGURE 1
Topics in Business Process Management

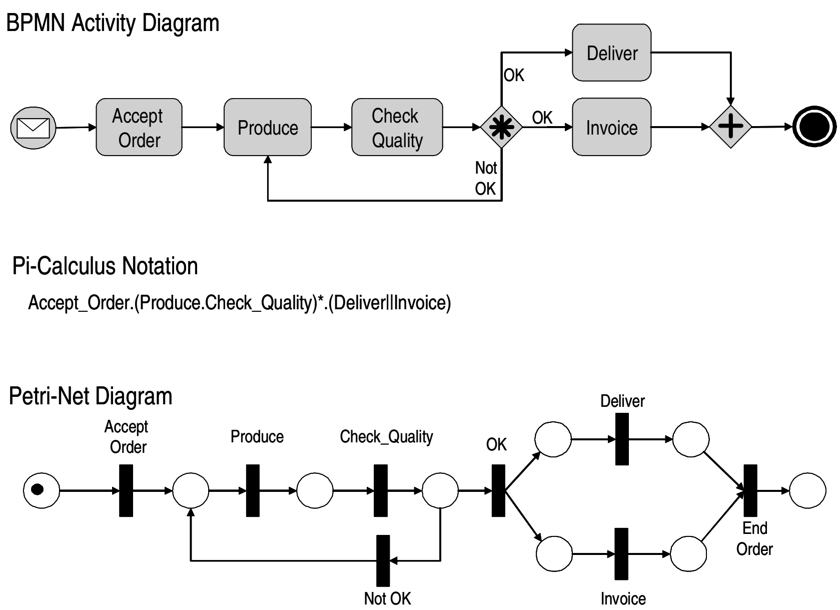


A. *Business Process Modelling*

The goal of business process modelling is to create a map of the business processes of a company. A typical characteristic of business processes is their hierarchical nature. A sales process can for example define that a sale consists of three consecutive steps: ordering, delivery and payment. Those three steps can at their turn be processes that can be refined at a lower level of decomposition. A complete process model addresses many different aspects, but traditionally, a major emphasis is put on the *process* aspects, being the order in which tasks are executed. From a technological perspective, BPMN

(Business Process Modelling Notation) is put forward as standard modelling technique. This technique is task-based: the main concepts of a BPMN diagram are the tasks which are interconnected by means of arrows to represent the sequence constraints. The BPMN notation shows a lot of resemblance with the Activity Diagrams as defined in UML (Unified Modelling Language). In the academic world, the preferred techniques are Petri-Nets and Process Algebra (π-calculus), mainly because of their formal mathematical nature. A major advantage of Petri-Nets is their graphical representation. A Petri-Net consists of places and transitions which are interconnected by means of an input/output relationship. A place represents the state of a particular case and a "token" represents a case in this particular state. A transition amounts to the execution of a task. A transition can only fire if there is a token in each input place and its firing will consume one token in each input place and produce one token in each output place. Figure 2 shows the equivalent schemas for a production process from the supplier's perspective according to the three modelling techniques (BPMN, π-calculus en Petri-Nets): after accepting an order, the supplier produces the requested product and subsequently checks whether it satisfies all quality criteria. If it doesn't, the production is restarted. If it does, the product is delivered and invoiced, delivery and invoicing being parallel activities. The individual tasks in the process can at their turn be exploded to new process definitions.

FIGURE 2
Modelling techniques for Business Processes



Ideally, business process modelling should not be restricted to the process aspects only, but should include many other aspects of the business, like required resources (information, people, machines, money …), timing aspects (duration, starting time, end time), authorisations, responsibilities, ownership, contribution to strategic goals, … and so on. Frameworks like ARIS (see Scheer (1999)) and ISA (see Zachman (1987)) list and organise the aspects to consider for a complete modelling of the enterprise, its processes and organisational aspects. The Zachman framework additionally includes the modelling of the supporting information systems.

B. *Business Process Analysis*

When modelling business processes, one can make a distinction between as-is and to-be models: an as-is model represents the current state of affairs whereas the to-be model represents the future desired state. Analysing the differences between the two models lays the ground for a change management plan.

In addition there is a broad spectrum of analysis techniques that help the modeller discover problems and opportunities for improvement in existing business processes. Smaller, gradual improvements are usually coined with the term *Business Process Improvement*, whereas a radical change of business processes is named *Business Process Reengineering*. Process analysis techniques are very diverse and depend on the technique used for modelling the business processes. Simulation is available for most of the modelling techniques. Formal techniques like Petri-Nets allow for certain types of well-formedness analysis such as liveness (is there always at least one transition that can fire ?) and boundedness (are there places where the number of tokens can grow above certain limits?). The combination with for example queuing theory yields additional possibilities for analysing quantitative aspects of processes such as throughput, turnaround time, and resource consumption.

C. *Business Process Enactment*

Once modelled, business processes should be put at work. This is the domain of *Business Process Enactment*. Enactment requires among other things the realisation of information systems offering users the required functionality to perform their tasks. As information systems embed numerous rules and constraints, very often process aspects such as the sequencing of tasks are hard coded into the information systems, for example by having attributes keep trace of the state of objects. However, one can obtain a much more flexible architecture by pushing the process aspects out of the information systems and having them managed by means of a business process engine. This requires formal business process definitions that are submitted as input to the business process engine. The BPM-engine will invoke the required information system services for the execution of a task, and will manage information flows between applications. Additional flexibility can be obtained by using a business rule engine to validate task execution against business rules. As the business process engine needs to interface with a variety of applications, technology from the area of *Enterprise Application Integration* (EAI) proves more than useful. Whereas modelling tools and EAI tools used to be separate domains, today tool builders try to gain advantage from the synergy between both domains. Modelling tools integrate with EAI tools to offer business process enactment functionality and EAI tools offer the possibility to define integration orchestration, for example by means of BPEL (Business Process Execution Language). The level of detail in BPEL is however quite low: a process is defined as the sequence of individual message exchanges between applications. Part of these message exchanges reflect the business process, but a substantial part of the messages reflects the fact that several applications need to be notified and collaborate for the execution of a single task. The BECO-model (see Snoeck et al. (2004), Lemahieu et al. (2005)) developed in our MIS research group aims at the separation of the business process aspects from the notification aspects in order to achieve an improved modularity, agility and maintainability of systems.

D. *Business Activity Monitoring*

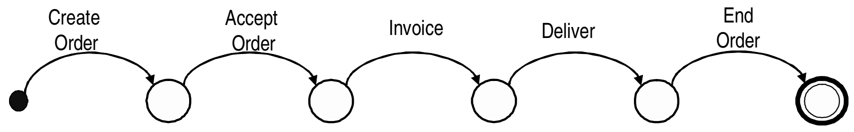
Business process enactment by means of a business process engine offers an interesting source of information for *Business Activity Monitoring*. For each case, the BPM-engine keeps track of the stage of the business process it is in: is the case completed or is it still in process with a particular user or application. The progress of business processes and cases can be inspected at run time: one can ask questions such as how many orders are ready for delivery, how many incoming payments still need to be processed. Looking post factum at completed processes also yields interesting information: how many orders are handled on average per day, what is the percentage of cancelled orders, what is the percentage of orders with a total amount smaller than 100 euro, … and so on. This type of information can also be obtained from data-warehouses and business intelligence tools, but the BPM form of monitoring offers the advantage of being always 100% up-to-date and of picturing the real-life business processes.

Introducing all aspects of business process management cannot be done at once, but should follow a staged approach. The maturity levels of the Capability Maturity Model (see Software Engineering Institute) provide an interesting evolutionary path to full business process management. Companies at the first level of maturity (Initial) are characterised by informal and ad hoc processes, which will be difficult if not impossible to capture in formal models. At the second level of maturity (Repeatable), similar tasks will be handled in similar ways. By capturing those repeatable processes in formal business process models, the company can reach the third level of maturity (Defined). By enacting those models and managing the business processes using activity monitoring, one can reach the fourth level of maturity (Quantitatively Managed). Finally, continuous Business Process Improvement brings a company to the highest level of maturity (Continuously Optimized).

E. *Challenges in Business Process Model Quality Control*

Because many tasks in business processes require some kind of support from an information system, business process models contain a lot of valuable information for information system developers. But at the same time, there is a substantial overlap between business process models and information system models. For example, UML allows modelling the life cycle of an object by means of a State Diagram. Figure 3 shows an example of such a diagram for the class ORDER. As a result, system developers will have to ensure that the sequence constraints imposed by the information system are compatible with the sequence constraints imposed by the business processes. We call this "*vertical compatibility"*. In the example there is no complete vertical compatibility: the information system imposes additional constraints compared to the business process model as it requires the invoicing to happen before the delivery, whereas this is not required by the business process model.

FIGURE 3
State Chart for Class ORDER



On the other hand, we also witness increasing e-business collaborations between companies, yielding cross-enterprise processes. Such cross-enterprise processes result from the composition of intra-company processes of the collaborating partners. *Horizontal compatibility* means that the processes of individual partners are compatible with each other with respect to the tasks for which they wish to collaborate. As an example, the ordering process of the customer could be as shown in Figure 4. Also in this case there is no complete compatibility: the customer reduces the scenarios of the supplier because his process defines that the delivery should precede the invoice. In addition, it is most annoying that this constraint is in contradiction with the life cycle of the class order.

FIGURE 4

The Customer's Ordering Process

This example illustrates that a thorough analysis of business processes and of the dynamic aspects of information systems is required to ensure the viability of business information systems. Horizontal and vertical compatibility must be ensured so as to allow the correct enactment of business processes. One of the topics on our research agenda is the development of Petri-Net based algorithms for the automatic verification of horizontal and vertical compatibility[1].

II. ENACTING BUSINESS PROCESSES AS WEB SERVICE ORCHESTRATIONS

One of the most promising technologies for enacting business processes are web services. This section first provides a basic overview of web services. Thereafter, we discuss the concept of *web service orchestration*, which allows for the tasks in a business process to be enacted through consecutive web service invocations.

A. *A basic overview of web services*

Wahli (2002) defines Web services as: "*self contained, self-describing, modular applications that can be published, located, and invoked across the Web*". Web services are coherent pieces of functionality that are published on the Web by a *service provider*. In principle, they can be implemented in any programming language, computing platform or operating system. A *service consumer* needn't be aware of the implementation details: a web service is invoked by simply sending an XML message to the appropriate web address. XML (eXtensible Markup Language) is a language to create documents or messages that contain structured information. XML serves many purposes, but a particular application of XML is the SOAP protocol, which specificies an XML format to interact with web services (see Seely and Sharkey (2001)). A SOAP message is a small XML document which represents an *operation invocation* and its corresponding *parameters*. The operation is executed by the web service to which the SOAP message was sent. The results of the execution are returned to the service consumer, again as an XML document.

In order to interact meaningfully with a web service, the service consumer needs to know the service's *interface*. This interface is defined by means of a WSDL document. Web Services Description Language or WSDL for short (see WSDL (2001)) is another XML based language, in which one can denote which operations can be invoked on a particular service and what the format of the SOAP message should be to invoke a particular operation. Each operation is described by means of an input SOAP message and (optionally) an output SOAP message.

In this way, the combination of SOAP and WSDL allows for ‘point-to-point’ integration between two applications or components, published as web services. Because SOAP can be used over internet protocols such as HTTP, it allows for a web service to be invoked from anywhere in the Web. Therefore, unlike other integration technologies, web services are not restricted to intra-enterprise application integration (EAI), but can also be used to integrate applications across an organization’s boundaries. The latter is called B2Bi: business-to-business integration.

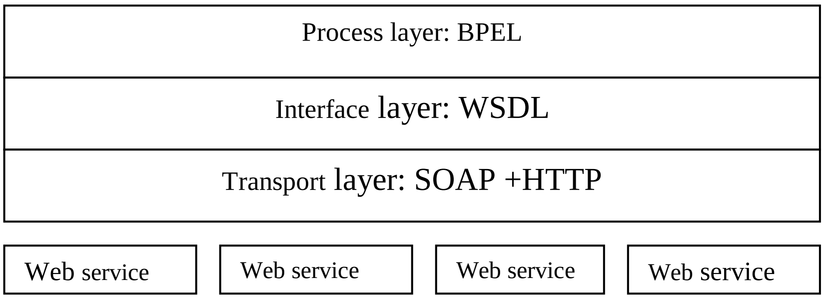
B. Web service orchestrations

The combination of SOAP and WSDL allows for a buyer’s web service to, for example, request the price of a certain product with the seller’s service. However, real business process enactment requires far more complex interactions than the simple point-to-point approach we described thus far. Such complex interactions can be modelled as web service *orchestrations*. A web service orchestration represents a business process as a consecution of subsequent (and sometimes parallel) tasks, with each task being realized as an operation invocation on a particular web service. Web service orchestrations can be specified by means of the BPEL language. BPEL (Business Process Execution Language, see Andrews (2003)) is, yet again, an XML based language. It allows specifying *executable processes*: each task in the process is tied to a web service invocation. Each such invocation corresponds to a SOAP message as specified in the corresponding service’s WSDL description. Whereas a WSDL document describes one particular services’ interface, the BPEL document describes how multiple services together perform the business process. It denotes which operations are to be called on which services, and in which sequence so as to perform a complex process. The BPEL document can be supplied to a BPM-engine in order to enact the corresponding process, hence the term *execution language*.

SOAP, WSDL and BPEL represent different layers in the web service technology stack. The latter is represented in Figure 5.

FIGURE 5

Layers in the web services technology stack



If the services in the BPEL document are offered by several independent companies, the orchestration specifies a business-to-business process. If all services belong to the same company, the BPEL document specifies a process internal to one company. A very popular example in a B2B context is that of a 'travelling agency' process. It denotes how the web services of a travelling agency, hotel chain, air carrier and car rental company should be orchestrated so as to allow a customer to book, confirm and pay an integrated package of a hotel booking, a flight and a rental car. In the next section, we discuss how electronic business is the most important application domain of web services and automated business process enactment, but we show a few other examples where very similar principles are applied in other domains as well.

III. APPLICATION DOMAINS OF BPM AND BUSINESS PROCESS ENACTMENT

A.. *Business-to-business integration and extended enterprises*

The first and most well known application of web services technology is in electronic business. A particular property of web services is that all standards involved are based on the universal XML language and the exchange of XML documents over existing Internet protocols. As a result, web services can easily be used as an integration technology across company borders. This is not possible with most other technologies, which are based on heavy-weight, proprietary protocols and are impossible to 'digest' by a companies' firewall.

Actually, Business-to-business integration comes in two flavors. *Market B2B* corresponds to a situation where independent companies dynamically find one another to participate in short lived, ad hoc transactions. Web services technology is not (yet) mature enough to provide fully automated support for this kind of interactions. A key problem is that this requires efficient web service registries, wherein potential service consumers can search and find the services they need. Such 'yellow pages' of web services already exist with the UDDI (Universal Description, Discovery and Integration) registry, but the latter lacks adequate and refined search capabilities to fully automate web service search and invocation.

Another form of B2Bi, which *is* already feasible with today's technology status, is *extended enterprise integration*. A so called extended enterprise consists of several independent companies, which together participate in a long term partnership. These fixed partners do not need search capabilities or registries to dynamically find one another. The independent partners integrate their information systems to a certain extent, in order to enact shared business processes and to reap the benefits of applying BPM at a level that surpasses an individual company. Rather than optimizing each separate link in a value chain that spans several companies, process analysis, management and improvement become possible at the level of the extended enterprise. Also, because web services facilitate automated collaboration, companies tend to concentrate on their core competences and outsource non-core activities to other partners.

When integrating business processes across several, independent organizations, a distinction is typically made between *public processes* and *private processes*. Public processes pertain to the interaction between the respective partners (customers, suppliers, shippers, …). Since the interaction takes places *between* organizations, there is a need for standardized protocols that are accepted and 'understood' by all parties. Web services suit this purpose nicely. Apart from that, each task in a public process may trigger one or more processes *internal* to the participating partners. These *private processes* contain activities which fall entirely within a single partner and which are often implemented by means of back-end applications such as ERP (Enterprise Resource Planning), SCM (Supply Chain Management) and CRM (Customer Relationship Management) systems. Since they are internal to a particular company, they do not require standardized web services technology and can be implemented by any approach 'proprietary' to that particular company. In this way, the web services can be seen as the boundary between public and private processes. Public processes are realized as orchestrations over web services, whereas a web service's underlying implementation may entail one or more private processes over a partner's back end applications. However, more and more, we see that private processes are realised as web service orchestrations as well.

Public processes are specified by means of *collaborative process models*, in which each partner plays a well defined *role*. BPEL offers facilities for specifying such collaborative processes. As already explained, a process consists of a consecution of tasks, with each individual task being realized as an operation invocation on a particular web service. In a collaborative process, each partner plays a specific role by acting as service provider for some services and acting as service consumer for services provided by other partners. For example, a purchase process will consist of a number of tasks which are to be executed in a well defined order, e.g. requesting a price, creating a purchase order, confirming the order, shipment, payment etc. Whereas one partner will be responsible for order processing, other tasks such as shipment, invoicing etc. may be delegated to other partners. Because BPEL offers a universal, XML based format, the process specification can be interpreted and processed (semi-)automatically by all partners' information systems.

Note that, in a collaborative process between several independent partners, every partner must 'agree' on the sequence of tasks to be performed in the process, as expressed in the BPEL document. This was called *horizontal compatibility* in section II. As a trivial example, if the supplier requires a *payment* task to take place before a *shipment* task, whereas the customer requires the opposite, both parties are incompatible process wise. One way to achieve compatibility is by using standard processes that are defined at the level of an entire industry branch. The most well know example of such standard is *RosettaNet* (see RosettaNet). The latter was specified by a consortium of high-tech companies (producers of consumer electronics, electronic components, semiconductors etc.) with as purpose the definition of a set of standards that facilitate automated business-to-business interaction. The most important part of RosettaNet are the PIPs (*Partner Interface Processes*). Whereas BPEL specifies a standard language for process definition, the PIPs define standards for the *processes themselves*. The PIPs include standards for requesting price and availability, querying order status, requesting shipment and much more. If all participating partners abide by these standards, they can be assured that they will be compatible process wise. Apart from (and less relevant than) the PIPs, RosettaNet also specifies an *implementation framework* which predates the web services specification, and which is gradually being replaced by the latter.

Another important standard with respect to process compatibility is ebXML, which was developed by the OASIS consortium (see ebXML). ebXML is (to a certain extent) compatible with RosettaNet PIPs, but adds the possibility of negotiation. A company can specify and publish its e-business capabilities in a so-called CPP (*Collaboration Protocol Profile*). Among these capabilities are the *process definitions* by which the company is able to do (on-line) business. When partners decide to collaborate, their CPPs are compared. Mismatches can be resolved in a semi-automated negotiation faze, the outcome of which results in a CPA (*Collaboration Protocol Agreement*). The CPA can be considered as a first attempt at an electronic 'contract', by which the collaborating partners agree to do business.

The concept of electronic contracts is currently lacking in the pure web services stack. The latter started out as being exclusively technology oriented, but the need for the integration of business related concepts becomes more and more apparent. It will become crucial to specify agreements, not only about the business process(es) to abide by, but also about other business-related issues such as the required quality of service, security requirements, process ownership, billing etc.

B. *Clinical pathways in the healthcare sector*

A noteworthy emerging application domain of BPM are Hospital Information Systems (see Liesmons (2004)). Most hospitals are still organized around a functional organization structure. Each department (medical, nursing, administrative, technical, ...) is an island of automation, with information being fragmented and possibly replicated in several supporting applications. The information exchange between the respective departments is often sporadic and hardly efficient. The different services administered by these departments are treated and optimized individually, without an eye for the care process as a whole.

However, more and more, national governments are pressuring hospitals to increase operational efficiency and cut down costs. A main factor is that hospitals are encouraged to reduce the number of hospital days, e.g. by the government stipulating a fixed rate for each diagnosis, regardless of the patient's length of stay in the hospital or the actual costs incurred during the stay. As a result, the length of stay turns into the main cost driver for hospitals, rather than the revenue driver it was before. In this context, process management becomes a very important issue in the healthcare sector, in order to guide a patient as swiftly and efficiently as possible through a set of tasks (medical tests and treatments, but also administrative activities), without dragging out the stay beyond what is strictly necessary.

More and more, we see an evolution to a process guided hospital organization, based around the patient and the health processes the latter is involved in. The trajectory followed by the patient, during which several tasks are performed, is called a clinical pathway. Clinical pathways are in fact the general concept of business processes applied to the care world. Again, processes can be designed, monitored and optimized, with overall objectives and quality criteria being established. Clinical pathways allow for a full, cohesive program to be applied to the patient, rather than individual, uncoordinated activities.

Hospitals that have instituted clinical pathways have seen substantial improvements in operational dimensions, but this approach also benefits the patient, by increasing the quality of the treatment. The process definition becomes a way of effectively coordinating all the different staff (skilled in very diverse disciplines, medical and non-medical) and making sure that the appropriate information is available at the right time for the right people. Each staff member is provided with a tailored view on the patient and on the current status of his personal clinical pathway, based on his admitting diagnosis. A further step is to extent this approach beyond the hospital's walls, for example to support home care.

C. *In-silico experiments in bioinformatics research*

Bioinformatics is a science discipline in which biology and information technology converge. It involves the development of algorithms and statistical techniques to analyze large data sets, in particular data sets about gene expressions and protein structures. This implies the development and implementation of tools that facilitate the analysis and the management of these heterogeneous types of data.

Bioinformatics research really took off with the advent of the Internet, which allowed for large interdisciplinary teams, distributed throughout the world, to collaborate on a specific problem. Initially, access to data and applications was achieved by cutting and pasting data in common web pages. However, this approach was inadequate to process large amounts of data or to perform complex experiments that involved several tools and data sets. More and more, bioinformatics experiments are executed by means of web services and web service orchestrations, in a manner very similar to e-business processes (see Greenwood et al. (2002)).

Indeed, in molecular biology, many experiments are based on combining and comparing new experiments with the results from previous studies. Both the tools to perform certain analyses and the data sets to perform these analyses on are being published as web services. Such so-called 'in-silico' experiments complement traditional 'in-vitro' experiments. On the one hand they allow for synthesizing new information from available data; on the other hand they yield new hypotheses, which can in their turn be confirmed by lab-based experiments. There is substantial use of electronic resources to compare new results with the existing body of biological knowledge. Discovery is done by combining and collating results obtained from a number of analysis tools and data resources. Each in-silico experiment applies one or more data manipulation and/or data analysis tools to one or more data sets. Both tools and data sets are published as web services. The experiment itself is described by a BPEL document, which specifies which manipulations should be applied to which data and in which sequence.

The fact that a scientist can represent his experiments as process definitions has several advantages. To begin with, it facilitates the reuse of (part of) the experimental setup, i.e. the experiments can easily be repeated with different data sets, different parameters etc. Equally important is that the process definition itself can be published. This allows for sharing of practices and the combination of (parts of) experiments by different parties. It also allows for defining 'process components': small orchestrations that are stored into a library and that can be reused and combined into a greater whole. Also, when one publishes the data that results from an experiment, the process definition that describes the experiment can be published with the data. This is very important for *data provenance*: the ability to track the origin of the data and the actual circumstances in which the data were obtained. Sometimes, this is required for legal purposes. However, an explicit record of which resources (data and tools) were used, which parameter settings where applied etc. also facilitates more accurate interpretation of the data. Finally, it is essential for assessing whether an in silico experiment needs to be re-run in the light of new evidence, such as updated resources, in order to meaningfully compare new results with existing data.

IV. A RESEARCH AGENDA FOR BPM

Business Process Management is a growing domain offering extensive opportunities for improved business-ICT alignment and efficiency. Nevertheless, to achieve its full potential, further research and improvement is required in a number of areas.

To begin with, business process models show a substantial overlap with information systems models and in particular with object oriented conceptual domain models: both address high level information modelling and the dynamic aspects of a business. In addition, there is a potential overlap in implementation issues as well: sequences between tasks can be modelled and implemented as part of a BPM system, but can just as well be enforced by the supporting information system. Moreover, business rule engines can be part of a BPM environment to manage different types of rules involved in business processes (e.g. domain constraints, derivation rules, triggers, …). Here again an overlap issue is raised: sequence constraints can be modelled and implemented as part of the process aspects but can also be enforced by introducing state attributes and having the business rule engine manage constraints on those attributes. As a result of these overlaps, care must be taken to ensure a clear separation of concerns. Further research should develop clear guidelines on the best way of separating concerns between business process, information system and business rule aspects.

Furthermore, the Internet offers challenging opportunities for inter-enterprise collaboration. This does however not always simplify organisational issues. As organisations grow more complex, so do their business process models, their set of business rules and their information system models. Because of the overlaps and strong links between business process models, business rules and information system models, verification of inter-model consistency is of crucial importance for achieving quality and optimal support in all areas. As systems and models grow in size and complexity, manual verification becomes unfeasible. The formal definition of modelling techniques and the development of companion verification techniques is hence an important research area. This research should aim at an improved integration between business process modelling and information system modelling, the pro-active assessment of morphological (e.g. liveness, boundedness) and quantitative (e.g. turn around time, throughput, resource consumption, ...) characteristics of business processes and at the development of (semi-) automated negotiation of collaboration protocols between e-business partners.

Finally, with web services becoming the primary technology for business process enactment, results from the above research issues should be applied to web service orchestration languages such as BPEL. Moreover, the abundance (and therefore lack of) standards for several aspects of web services (this paper discussed only the tip of the iceberg) should be resolved. In this standardization process, a balance should be stricken between technological issues and business related issues.

This research agenda proposed a few points of attention, but is in no way exhaustive. A lot has been realised since the early days of workflow management, but many open issues remain. Because BPM exists at the very boundary between business and information systems, the management informatics research discipline is very well placed to look into them.

NOTES

1. Part of this research is funded by the Microsoft Research Chair.

REFERENCES

Andrews, T. et al., 2003, Specification: Business Process Execution Language for Web Services Version 1.1, ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf.

ebXML, www.ebxml.org.

Greenwood, M., Wroe, C., Stevens, R., Goble, C. and Addis, M., 2002, Position Paper: Are Bioinformaticians doing e-Business?, Proceedings of EuroWeb 2002, (Oxford, UK).

Lemahieu, W., Snoeck, M., Goethals, F., De Backer, M., Haesen, R., Dedene, G. and Vandenbulcke, J., 2005, Coordinating COTS Applications via a Business Event Layer, *IEEE Software* 22, 4.

Liesmons, I., 2004, The Added Value of a Process Oriented Hospital Information System Supporting the Integrated Patient Care, Studies in Health Technology and Informatics Volume 110: Health Continuum and Data Exchange in Belgium and in the Netherlands.

RosettaNet, www.rosettanet.org.

Scheer, A.-W., 1999, ARIS : Business Process Modeling, (Springer, Berlin).

Seely, S. and Sharkey, K., 2001, SOAP: Cross Platform Web Services Development Using XML, (Prentice Hall, Upper Saddle River, NJ).

Snoeck, M., Lemahieu, W., Goethals, F., Dedene, G. and Vandenbulcke, J., 2004, Events as Atomic Contracts for Application Integration, *Data and Knowledge Engineering* 51 81-107.

Software Engineering Institute, Capability Maturity Models, www.sei.cmu.edu/cmm/cmms/cmms.html.

Wahli, U., 2002, Self-Study Guide: WebSphere Studio Application Developer and Web Services, *IBM Redbook*, www.redbooks.ibm.com/redbooks/pdfs/sg246407.pdf.

Web Services Description Language (WSDL) 1.1. Specification, 2001, www.w3.org/TR/wsdl.

Zachman, J., 1987, A Framework for Information Systems Architecture, *IBM Systems Journal* 26, 3, 276-292.