5

Business Process Data Analysis

The problem of understanding the behavior of information systems as well as the processes and services they support has become a priority in medium and large enterprises. This is demonstrated by the proliferation of tools for the analysis of process executions, system interactions, and system dependencies, and by recent research work in process data warehousing and process discovery. Indeed, the adoption of business process intelligence techniques for process improvement is the primary concern for medium and large companies. In this context, identifying business needs and determining solutions to business problems requires the analysis of business process data. Analysis of business data will help in discovering useful information, suggesting conclusions, and supporting decision making for enterprises, and enable process analyst to answer questions such as: *where are the bottlenecks in the purchasing process? what is the actual process typically followed for invoice payment? what is the status of purchase order number 325, who processed it and how?, and how to find all the information related to a specific purchase of a customer (order, invoice, payment, shipping, etc) in the enterprise?*

In this chapter, we give an overview of different aspects of business data analysis techniques and approaches from process/data spaces to data provenance and data-based querying techniques. We start with an overview of *warehousing process data* follow by introducing *data services* and *DataSpaces* which facilitate organizing and analyzing process related data. Next we discuss the importance of supporting big data analytics over process execution data. Afterward we define a holistic view of the process executions over various information systems and services (i.e. Process Space) following by a brief overview of process mining to highlight the interpretation of the information in the enterprise in the context of process mining. Finally, we focus on process artifacts and introduce cross-cutting aspects in processes data

and discuss how process analytics can benefit from cross-cutting aspects such as provenance, e.g. to analyze the evolution of business artifacts.

5.1 Warehousing Business Process Data

Improving business processes is critical to any corporation. Process improvement requires analysis as its first basic step. Before analyzing the process data, there is a need to capture and organize the process data. This is important as executions of process steps, in modern enterprises, leave temporary/permanent traces in various systems and organizations. In order to analyze process data it is possible to collect the data into a data warehouse, using extract, transform, load (ETL) tools, and then leverage an OLAP tool to slice and dice data along different dimensions [105].

In this context, the process data warehousing presents interesting challenges [105]: (i) outsourcing: developing an ad-hoc and process-specific solutions for warehousing and reporting on process data is not a sustainable model; (ii) process data abstraction: the typical process executed in the IT system is very detailed and consists of dozens of steps, including manual operations (e.g., scanning invoices), database transactions, and application invocations; (iii) data evolution: the business process automation/analysis application are co-developed, which means that, during development, changes to the data sources and even to the reporting requirements are fairly frequent.

Considering the above mentioned challenges, and in the domain of business processes, it is important to devise a method for minimizing the impact of changes and be able to quickly modify and re-test the ETL (extract, transform, and load) procedures, the warehouse model, and the reports. To address these challenges, Casati et al. [105] proposed a conceptual model for process data warehousing. In particular, They provided a configurable warehouse model that can satisfy complex reporting needs for virtually any process, also taking into account performance constraints. The model addresses key recurring problems such as the trade-off between the need to model heterogeneity (each process is different) and that of defining a uniform representation for all processes (to support reusability and cross-process analysis).

To support warehousing for business process data, there is a need to provide users with a way to model the abstraction. This will help understanding the high level processes and also it will describe how its progression maps to underlying IT events. Moreover, there is a need for an ETL mechanism that, based on the abstract process definition and the events occurring on the different systems, loads the warehouse with abstracted process execution data. To address these requirement, modeling abstract processes should involve: (i) describing the process flow; (ii) specifying how the abstracted business data for each process is populated and maintained; (iii) associating the start and completion of each step with changes to the abstract business data; and (iv) associating steps to human or automated resources.

To populate the process data warehouse it is necessary to first extract the data from the different event log databases into the landing tables of the staging area.

In the process domain, data services play an important role in extracting process related data. Following we explain data services and we discuss how they provide an emulation environment that supports testing and prototyping of events and data: once a process is started, each step binds to a data service and will be assigned for execution to a data generation web service.

5.1.1 Data Services

In the enterprise world, data services play an important role in SOA architectures [102, 103, 207, 402]. For example, when an enterprise wishes to controllably share data (e.g., structured data such as relational tables, semi-structured information such as XML documents, and unstructured information such as commercial data from online business sources) with its business partners, via the Internet, it can use data services to provide mechanisms to find out which data can be accessed, what are the semantics of the data, and how the data can be integrated from multiple enterprises. In particular, data services are "software components that address these issues by providing rich metadata, expressive languages, and APIs for service consumers to send queries and receive data from service providers" [103].

A Web service, i.e., a method of communication between two electronic devices over the Web [32], can be specialized, as a data service, to encapsulate a wide range of data-centric operations, where these operations need to offer a semantically richer view of their underlying data in order to use or integrate entities returned by different data services [103]. Microsoft's WCF data-services framework[1], which enables the creation and consumption of OData services for the web, and Oracle's ODSI[2], which provides a wide array of data services designed to improve data access from disparate data sources for a wide range of clients, are two of a number of commercial frameworks that can be used to achieve this goal.

In this context, SOA applications will often need to invoke a service to obtain data, operate locally on that data, and then notify the service of changes that the application wishes to make to the data. Consequently, standards activity is needed in the context of data services. For example, the Open SOA Collaborations Service Data Objects (SDO) specification [402] addresses these needs by defining client-side programming models, e.g., for operating on data retrieved from a data service and for XML serializing objects, and their changes for transmission back to a data service [102]. In particular, the use of data is bound to various rules imposed by data owners and the (data) consumers should be able to find and select relevant data services as well as utilize the data 'as a service'.

Data as a service, or DaaS, is based on the concept that the data can be provided on demand to the user regardless of geographic or organizational separation of provider and consumer [448]. In particular, data services are created to integrate

[1] http://msdn.microsoft.com/en-us/data/bb931106

[2] http://docs.oracle.com/cd/E13162_01/odsi/docs10gr3/

as well as to service-enable a collection of data sources. These services can be used in mashups, i.e., Web applications that are developed starting from contents and services available online, to use and combine data from two or more sources to create new services. In particular, data services will be integral for designing, building, and maintaining SOA applications [102]. For example, Oracle's ODSI supports the creation and publishing of collections of interrelated data services, similar to *dataspaces*.

Data services can be leveraged to reduce the effort required to set up a data integration system and to improve the system in 'pay-as-you-go' fashion as it is used. In this context, data integration approaches require semantic integration before any services can be provided. It is important as process data is scattered across several systems and data sources and there is no single schema to which all the process related data conforms. To address this challenge, *Dataspaces* proposed to overcome some of the problems encountered in data integration system and to promote awareness of the data and address concerns for ensuring the long-term availability of data in repositories.

5.1.2 DataSpaces

Dataspaces are an abstraction in data management that aim to manage large number of diverse interrelated data sources in enterprises in a convenient, integrated, and principled fashion. Dataspaces are different from data integration approaches in a way that they provide base functionality over all data sources, regardless of how integrated they are. For example, a dataspace can provide keyword search over its data sources, then more sophisticated operations (e.g., mining and monitoring certain sources) can be applied to queried sources in an incremental, pay-as-you-go fashion [210]. These approaches does not consider the business process aspects per se, however, they can be leveraged for organizing and managing ad-hoc process data.

DataSpace Support Platforms (DSSPs), have been introduced as a key agenda for the data management field and to provide data integration and querying capabilities on (semi-)structured data sources in an enterprise [210, 415]. For example, SEMEX [98] and Haystack [259] systems extract personal information from desktop data sources into a repository and represent that information as a graph structure where nodes denote personal data objects and edges denote relationships among them.

The design and development of DSSPs have been proposed in [173]. In particular, a DSSP [173, 210, 415]:

- helps to identify sources in a dataspace and inter-related identified resources. A DSSP is required to support all the data in the dataspace rather than leaving some out, as with DBMSs;
- offers basic searching, querying, updating, and administering mechanisms over resources in a dataspace, including the ability to introspect about the contents.

However, unlike a DBMS, a DSSP is not in full control of its data, as same data may also be accessible and modifiable through an interface native to the system hosting the data;

- does not require full semantic integration of the sources in order to provide useful services: there is not a single schema to which all the data conforms and the data resides in a multitude of host systems;
- offers a suite of interrelated (data integration and querying) services in order to enable developers focusing on the specific challenges of their applications. Queries to a DSSP may offer varying levels of service, as sometimes individual data sources are unavailable and best-effort or approximate answers can be produced at the time of the query;
- provides mechanisms for enforcing constraints and some limited notions of consistency and recovery, i.e., to create tighter integration of data in the space as necessary.

Motivating applications for DSSPs includes scenarios in which related data are scattered across several systems and data sources, e.g., personal information management systems [157], which are used to acquire, organize, maintain, retrieve and use information items (e.g., desktop documents, web pages and email messages) accessed during a person's lifetime, and scientific data management systems [196], which are used for record management for most types of analytical data and documentation which ensures long-term data preservation, accessibility, and retrieval during a scientific process.

In order to search and query dataspaces, a new formal model of queries and answers should be specified. This is challenging as answers will come from multiple sources and will be in different data models and schemas. Moreover, unlike traditional querying/answering systems, a DSSP can also return sources, i.e., pointers to places where additional answers can be found. Some works [211, 329] presented semantic mappings techniques to reformulate queries from one schema to another in data integration systems. Another line of related work [80, 206] focused on ranking answers in the context of keyword queries to handle the heterogeneity of resources. Some other works, e.g., in [294], focused on finding relevant information sources in large collections of formally described sources.

In dataspaces, a significant challenge is to answer historical queries which applied to heterogenous data. A line of research proposed techniques for modeling and analyzing provenance [116] (also known as lineage and pedigree), uncertainty and inconsistency of the heterogenous data in dataspaces [244, 472]. Many provenance models [116, 175, 348, 423] have been presented, motivated by notions such as influence, dependence, and causality in such systems. Moreover, the relationship between uncertainty and provenance discussed in [472].

Dataspaces are large collections of heterogeneous and partially unstructured data, and therefore, indexing support for queries that combine keywords and the structure of the data can be challenging. For example, in [149], authors proposed an indexing technique for dataspaces to capture both text values and structural information using an extended inverted list. Their proposed framework extend inverted lists that capture attribute information and associations between data items, i.e., to support robust

indexing of loosely-coupled collections of data in the presence of varying degrees of heterogeneity in schema and data. Another indexing system [138], designed to provide entity search capabilities over datasets as large as the entire 'Web of Data'. Their approach supports full-text search, semi-structural queries and top-k query results while exhibiting a concise index and efficient incremental updates. Challenges in implementing a scalable and high performance system for searching semi-structured data objects over a large heterogeneous and decentralized infrastructure have been discussed in [137], where an indexing methodology for semi-structured data have been introduced.

Recently, new class of data services designed for providing data management in the cloud [460]: the cloud is quickly becoming a new universal platform for data storage and management. In practice, data warehousing, partitioning and replication are well-known strategies to achieve the availability, scalability, and performance improvement goals in the distributed data management world. Moreover, database-as-a-service proposed as an emerging paradigm for data management in which a third party service provider hosts a database as a service [207]. Data services can be employed on top of such cloud-based storage systems to address challenges such as availability, scalability, elasticity, load balancing, fault tolerance, and heterogenous environments in data services. For example, Amazon Simple Storage Service (S3) is an online public storage Web service offered by Amazon Web Services[3].

A growing number of organizations have begun turning to various types of non-relational, *NoSQL* (not only SQL), databases such as Google Bigtable [110], Yahoo PNUTS [127], and Amazon Dynamo [136]. NoSQL is a broad class of low-cost and high performance database management systems and proposed to address RDBMSs shortcomings: ever-increasing needs for scalability and new advances in Web technology, which requires facilitating the implementation of applications as a distributed and scalable services, have created new challenges for RDBMSs [430, 460]. Such databases are designed to be very scalable and reliable and they consists of thousands of servers geographically distributed all over the world. Major research challenges for providing heterogenous data management, e.g. using data services, need [102, 103, 207]: (i) a dynamically reconfigurable runtime architectures, distributed service components and resources should be leveraged to create an optimal architectural configuration to both a particular users requirements and the application characteristics; (ii) an end-to-end security solutions, a full system approach to test end-to-end security solutions at both the network and application level is required; (iii) the infrastructure support for data and process integration, uniform consistent access to all heterogenous data should be provided, i.e., irrespective of the data format, source, or location; and (iv) the analytic support for the discovery and communication of meaningful patterns in (process execution) data, e.g., *business analytics*.

The field of business analytics has improved significantly over the past few years, giving business users better insights, particularly from operational data managed by dataspaces. For example, banks that developed an analytic application for budgeting

[3] http://aws.amazon.com/

and forecasting targeted at the financial services industry determined that its online analytical processing, or OLAP, can provide the capability for complex calculations, trend analysis, and sophisticated data modeling. In particular, OLAP can be used to reduce the time needed for analyzing the process data by providing powerful user-interfaces that let the analyst explore the process related data along previously defined analysis dimensions.

5.2 Supporting Big Data Analytics Over Process Execution Data

In modern enterprises, businesses accumulate massive amounts of data from a variety of sources. In order to understand businesses, one needs to perform considerable analytics over large hybrid collections of heterogeneous and partially unstructured process related execution data. This data increasingly come to show all typical properties of the *big data*: wide physical distribution, diversity of formats, non-standard data models, independently-managed and heterogeneous semantics and needs to be represented as graphs, i.e. *big process graphs*. The discovery and communication of meaningful patterns in data (i.e. analytics) can help in understanding the big business data with an eye to predicting and improving business performance in the future.

In order to understand available data (events, business artifacts, data records in databases, etc.) in the context of process execution, we need to represent them, understand their relationships and enable the analysis of those relationships from the process execution perspective. To achieve this, it is possible to represent process-related data as entities and any relationships among them (e.g., events relationships in process logs with artifacts, etc.) in entity-relationship graphs. In this context, business analytics can facilitate the analysis of process graphs in a detailed and intelligent way through describing the applications of analysis, data, and systematic reasoning [41, 97, 224, 272]. Consequently, an analyst can gather more complete insights using techniques such as modeling, summarizing, and filtering.

Applications of business analytics extend to nearly all managerial functions in organizations. For example, considering financial services, applying business analytics on customer dossiers and financial reports can specify the performance of the company over periods of time. As another example, consider the collaborative relationship between researchers, affiliated with various organizations, in the process of writing scientific papers, where it would be interesting to analyze the collaboration-patterns [29, 31] (e.g., frequency of collaboration, degree of collaboration, mutual impact, and degree of contribution) among authors or analyze the reputation of a book, an author, or a publisher in a specific year. Such operations, requires supporting n-dimensional computations on process graphs, providing multiple views at different granularities, and analyzing set of dimensions coming from the entities and the relationship among them in process graphs.

In traditional databases (e.g., relational DBs), data warehouses, OLTP (Online Transaction Processing) and OLAP (On-Line Analytical Processing) technolo-

gies [18, 111] were conceived to support decision making and multidimensional analysis within organizations. To achieve this, a plethora of OLAP algorithms and tools have been proposed for integrating data, extracting relevant knowledge, and fast analysis of shared business information from a multidimensional point of view. Moreover, several approaches have been presented to support the multidimensional design of a data warehouse. Cubes defined as set of partitions, organized to provide a multi-dimensional and multi-level view, where partitions considered as the unit of granularity. Dimensions defined as perspectives used for looking at the data within constructed partitions. Furthermore, OLAP operations have been presented for describing computations on cells, i.e. data rows.

While existing analytics solutions, e.g., OLAP techniques and tools, do a great job in collecting data and providing answers on known questions, key business insights remain hidden in the interactions among objects and data: most objects and data in the process graphs are interconnected, forming complex, heterogeneous but often semi-structured networks. Traditional OLAP technologies were conceived to support multidimensional analysis, however, they cannot recognize patterns among process graph entities and analyzing multidimensional graph data, from multiple perspectives and granularities, may become complex and cumbersome. Existing approaches [59, 115, 167, 221, 255, 269, 391], in on-line analytical processing on graphs, took the first step by supporting multi-dimensional and multi-level queries on graphs, however, much work needs to be done to make OLAP heterogeneous networks a reality [220]. The major challenges here are: (i) how to extend decision support on multidimensional networks, e.g., process graphs, considering both data objects and the relationships among them; and (ii) providing multiple views at different granularities is subjective: depends on the perspective of OLAP analysts how to partition graphs and apply further operations on top of them.

Besides the need to extend decision support on multidimensional network in process data analysis scenarios, the other challenge is the need for scalable analysis techniques. Similar to scalable data processing platforms [483], such analysis and querying methods should offer automatic parallelization and distribution of large-scale computations, combined with techniques that achieve high performance on large clusters, e.g. cloud-based infrastructure, and be designed to meet the challenges of process data representation that should capture the relationships among data (mainly, represented as graphs). In particular, there is a need for new scalable and process-aware methods for querying, exploration and analysis of process data in the enterprise because: (i) process data analysis methods should be capable of processing and querying large amount of data effectively and efficiently, and therefore have to be able to scale well with the infrastructure's scale; and (ii) the querying methods need to enable users to express their data analysis and querying needs using process-aware abstractions rather than other lower level abstractions.

To address these challenges, P-OLAP [57] (Process OLAP) proposed to support scalable graph-based OLAP analytics over process execution data. The P-OLAP goal is to facilitate the analytics over big process graph through summarizing the process graph and providing multiple views at different granularity. P-OLAP, benefits from BP-SPARQL [65] (Business Process SPARQL), a MapReduce-based graph

processing engine, for supporting big data analytics over process execution data. P-OLAP framework have been integrated into ProcessAtlas [65], process data analytics platform, which introduces a scalable architecture for querying, exploration and analysis of large process data.

5.2.1 OLAP (On-Line Analytical Processing)

There is an analytical orientation to the nature of the process data, and consequently process analytics can benefit from decision support systems and business intelligence tools. Consequently, process analytics can benefit from Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP) to reduce the time needed for analyzing the process data. This is important as the large amount of process related data generated every second need to be analyzed in almost realtime. OLTP proposed to facilitate and manage transaction-oriented applications, typically for data entry and retrieval transaction processing. OLAP proposed to support analysis and mining of long data horizons, and to provide decision makers with a platform from which to generate decision making information. In this section we focus on OLAP environments and discuss its importance in process analytics.

OLAP applications typically access large (traditional) databases using heavyweight read-intensive queries. OLAP encompasses *data decision support* (focusing on interactively analyzing multidimensional data from multiple perspectives) and *data mining* (focusing on computational complexity problems). There have been a lot of works, discussed in a recent survey [404] and a book [444], dealing with multidimensional modeling methodologies for OLAP systems. Multidimensional conceptual views allow OLAP analysts to easily understand and analyze data in terms of facts (the subjects of analysis) and dimensions showing the different points of view where a subject can be analyzed from. These line of works, propose OLAP data elements such as partitions, dimensions, and measures and their classification, e.g., classifying OLAP measures into distributive, algebraic and holistic. They discuss that one fact and several dimensions to analyze it give rise to what is known as the data cube.

There are many works, e.g., [190, 219], dealing with the efficient computation of OLAP data cubes, including: (i) efficient methods for computing iceberg-cubes[4] [219] with some popularly used measures, such as average; (ii) efficiently compute multiple related skyline[5] results; (iii) compute closed iceberg-cubes more efficiently using aggregation-based approach; and (iv) proposing an algebra that operates over data cubes, independently of the underlying data types and physical data representation [190].

---

[4] An Iceberg-Cube contains only those cells of the data cube that meet an aggregate condition. It is called an Iceberg-Cube because it contains only some of the cells of the full cube [79].

[5] Skyline [478] has been proposed as an important operator for multi-criteria decision making, data mining and visualization, and user preference queries.

Many other works, e.g., [26, 178, 303], deal with clustering and partitioning of large databases, including: (i) presenting a classification of OLAP queries to decide whether and how a query should be parallelized [26]; (ii) proposing an efficient solution, called adaptive virtual partitioning (AVP), for parallel query processing in a database cluster [303]; (iii) combining the physical and virtual partitioning to define table subsets in order to provide flexibility in intra-query parallelism; (iv) analyzing independent data tuples that mathematically form a set, i.e. conventional spreadsheet data; (v) leveraging both spreadsheets and ad-hoc OLAP tools to assess the effects of hypothetical scenarios [46]; and (vi) clustering and classification of graphs by studying systematically the methods for mining information networks and classifying graphs into a certain number of categories by similarity [220]. All these works provide some kind of (network) summaries incorporates OLAP-style functionalities.

Other works [57, 60, 156, 176, 391] focused on mining and querying information networks, including: (i) proposing techniques for query processing and cube materialization on informational networks [391]; (ii) defining constraints on nodes and edges simultaneously on the entire object of interest, not in an iterative one-node-at-a-time manner. Therefore, they do not support querying nodes at higher levels of abstraction [156, 176]; (iii) proposing summarization frameworks to facilitate the analysis of process data modeled as graphs [60]; and (iv) facilitating the analytics over big process graph, P-OLAP [57], through summarizing the process graph and providing multiple views at different granularity.

5.2.2 Trend, What-If and Advanced Analysis

Various methods and techniques have been proposed for analysis and interpretation of process data. The focus of these techniques is on the behavior of completed processes, evaluate currently running process instances, and predicting the behavior of process instances in the future. Some of these techniques [5, 10, 54, 345] are purely syntax oriented, focusing on filtering, translating, interpreting, and modifying event logs given a particular question. Other methods [91, 104, 230, 331] focused on the semantics of process data and tried to propose techniques to understand the hidden relationships among process artifacts. In particular, existing works on business analytics focused more on exploration of new knowledge and investigative analysis using broad range of analysis capabilities, including: trend analysis, what-if analysis, and advanced analysis.

The focus in trend analysis is to explore data and track business developments with capabilities for tracking patterns. For example, it is possible to track patterns in Web services, as services leave trails in so-called event logs and recent breakthroughs in process mining research make it possible to discover, analyze, and improve business processes based on such logs [7]. Also it is possible to monitor the status of running processes and trace the progress of execution [54, 345], or enabling semantic process mining in order to track business patterns [91, 104, 230, 331].

Some of these techniques, e.g., [331] and [91], implemented as plugins in the ProM framework tool.

In what-if analysis, scenarios with capabilities for reorganizing, reshaping and recalculating data is of high interest. In this category, business process data can be used to forecast the future behavior of the organization through techniques such as scenario planning and simulation [112]. One research direction is to explore the relationships between what-if analysis and multidimensional modeling [276], and to analyze the natural coupling, which exists between data modeling, symbolic modeling and what-if analysis phases of a decision support systems. Another line of research describes what-if analysis as a data intensive simulation to inspect the behavior of a complex system under some given hypotheses [189]. Also it is possible to perform what-if analysis, by investigating the requirements for a process simulation environment.

Advanced analysis techniques, provide techniques to uncover patterns in businesses and discover relationships among important elements in an organization's environment. Linking between entities across repositories has been the focus of a large number of works. For example, the idea of Linked-Data[6] has recently attracted a lot of attention in information systems, where research directions include: discovery of semantic links from data based on declarative specification of linkage requirements by a user [224], link semantically related entities across internal and external data sources using the power of external knowledge bases available on the Web [225], investigating the problem of event correlation for business processes [352] to detect correlation identifiers from arbitrary data sources in order to determine relationships between business data [407], and to discovering inter-process relationships in a process repository [284].

Moreover, a new stream of work [95, 192, 325] has focused on weaving social technologies to business process management. They aim to consolidate the opportunities for integrating social technologies into the different stages of the business process lifecycle, in order to discover the hidden relationships among process artifacts. Social-BPM takes advantage of social media tools, e.g. enriching business processes life-cycle with tagging information [314], to improve communication. A novel research direction in advanced analysis techniques could be using Natural Language Processing (NLP[7]) and Coreference Resolution [64] (CR) techniques to analyze the process related documents and to discover more insight from hidden information in the text documents.

5.3 Business Data Analytics and Process Spaces

Existing business process management tools enable monitoring and analysis of *operational* business processes, i.e., the ones that are explicitly defined and the process

[6] Linked Data is a method of publishing data on the web based on principles that significantly enhance the adaptability and usability of data, either by humans or machines [87].

[7] http://nlp.stanford.edu/

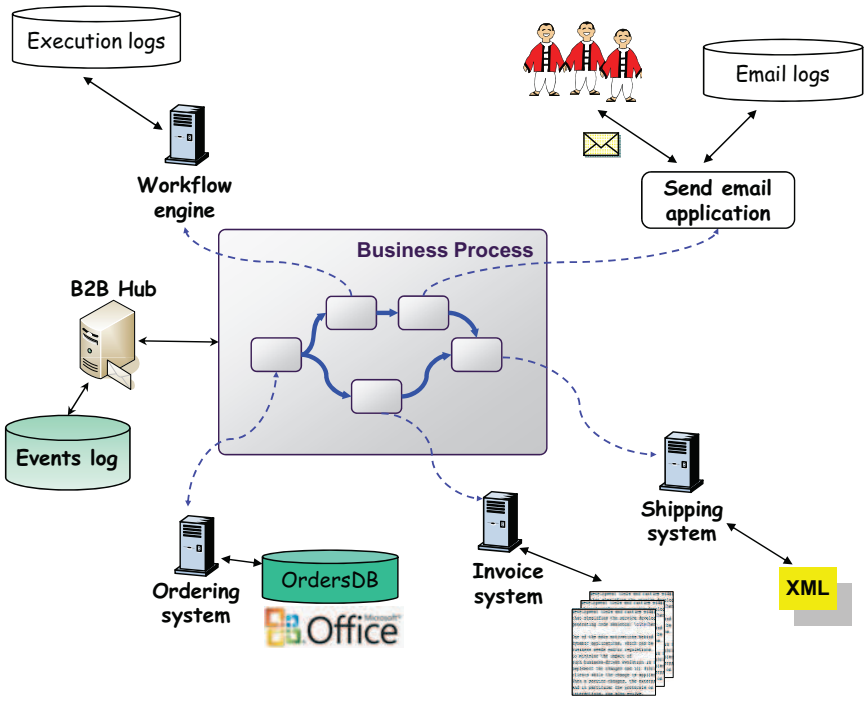

Fig. 5.1 An example of the business process execution in modern enterprises.

is managed by a process-aware system, such as a workflow management system (WfMS) [11, 184, 295, 453]. However, in reality, only fraction of process executions is supported by a WfMS and business process is implemented across several heterogeneous IT systems. Gartner identifies process analysis and monitoring in such environments as a major challenge that play a vital role in the survival and competitiveness of process management systems vendors in near future [179].

As a motivating example, consider the processing of purchase orders in a typical enterprise depicted in Figure 5.1: the order is first received by the company through a B2B hub, which can be either a in-house developed Web service, an EDI receptor, or an e-commerce application such as Ariba, which begin with logging the related events and order verification. Once verified, the order is routed to the purchase order management system that initiates the approval process. Once approved, the processing of the order may require interaction with the workflow system, for procurement, but is also inevitably characterized by email and document exchanges among people as part of processing. At the same time, a notification may be sent to the invoice and payment systems, and finally the shipping system to arrange for the shipping. During this process, documents (e.g., the purchase order and approval documents) may

be stored in a document management system (e.g., a Microsoft Sharepoint server) to facilitate collaborative editing and viewing of documents.

The above description shows that process execution information ends up being composed of a sea of apparently uncorrelated information items scattered across various systems in the enterprise. In fact, the situations where data on the entire process is conveniently located into one system and format do not happen in reality. Therefore, we need to analyze process data and understand the origin of data (i.e. *Provenance* [116]) in the enterprise so that they can be seen "as if" they were captured by a single business process management system. This implies to define a holistic view of the process executions over various information systems and services (i.e. Process Space) and to enable interpretation of the information in the enterprise in the context of process mining.

5.3.1 Process Space

A process space is essentially composed of (i) definition of criteria or mechanisms for deciding which information items in the enterprise are *correlated*, i.e., belong to the same execution (instance) of a process, (ii) a way of *mapping* information items to process progression events (start and completion of process tasks), and (iii) a *process models* of the processes in the enterprise. In a process space, different process models, mappings, and correlations, can be defined over the same set of information items as different analysts may be interested in different views over such events (called as *process views*). For example, the shipments of a set of goods may be related from the view of the warehouse manager, but if the goods are the results of different orders, they are unrelated from the view of the sales manager. As an example scenario, consider ACME enterprise, depicted in Figure 5.1, which supplies semiconductors to a variety of manufacturers in markets.

Example 1: Search and query process spaces. A business manager, Amy, receives a complaint email about a purchase order number 325 from a manufacturer. To properly respond to the complaint, she wants to find all the information on the process execution from the order request to the order completion, including product order details, people involved in the order approval process, emails and documents exchanged among them, payment and billing information, shipping information, etc. For instance these include email ABC, shipping document PO325.xml and invoice xyz.pdf). Currently, Amy has to *individually* search each data sources of the enterprise and look for information of the order number 325. For instance, he first need to search the order order DB, and get the invoice number from there. Then she has to search the invoice system to find related information and also payments, and shipping subsequently going through emails and documents stored in various systems. This is a very daunting and time consuming task.

A dataspace management system [173], which aims at making it possible to access data from multiple and heterogeneous data sources, does not provide the ability to interpret the data in the context of process executions. Other available

techniques and technologies, e.g., enterprise information integration and enterprise search tools [209] also do not provide ways to find information regarding process executions in the enterprise. What is needed here is a kind of a process space management system that can offer the functionalities required in this scenario including: (i) allowing to browse all the information about process executions across data sources, and to identify relationships among information in terms of process execution, (ii) enabling to index a large variety of information including business documents (e.g., as Microsoft word, emails, tuples in DB and XML documents) along with process execution context (e.g., events or process instances) in order to support efficient search and query of process executions.

Example 2: Monitoring and analyzing process spaces. An IT manager, Bob, is interested in monitoring and analyzing process executions to prevent similar complaints from happening in the future. He may ask which purchase order cannot be completed at the scheduled date, where the bottlenecks in the process are, and who has been working on the order. In addition, he wants critical situations to be identified and alerted before such situations arise, e.g., when there is a high probability that the shipping phase is not going to finish on time [105]. With current technologies such as WfMS and business process intelligence (BPI) techniques [201], it is difficult to monitor and analyze process executions supported by one or more systems (web sites, databases, document management systems, ERP systems, message brokers, workflow systems, etc), which not all of them are process-aware.

Current process monitoring and analysis approaches (e.g., [54, 55]) also require a process model of the process to be analyzed. Hence, this limits the analysis to workflow log meaning that we can only monitor and analyze a fraction of processes in an enterprise. An important enabler of process analysis across such systems is the ability to correlate events and documents and discover process views for various users including Bob. What is missing is such a system so to enable performing analysis tasks on most or all sorts of processes not only those supported by the workflow systems. For example, the approach proposed in [201, 202], considered the existence of a process model and focused on mining process execution data using decision trees to find interesting correlations between process data and behaviors (e.g. delays, outcomes, and more generally, behavior patterns defined by analysts).

5.3.2 Logical components of process spaces

Business processes in an enterprise are implemented using various (heterogeneous) information systems and services. A *process space* is the set of data sources containing information related to the execution of processes in the enterprise over which we superimpose a business process metaphor. Some examples of process spaces are purchase order process space, insurance claim process space, mortgage application process space, and auction process space. The data sources in the process space can be categorized into two types:

- **Events data sources** which refer to data sources that store meta-data about the *events* related to the execution of business process and exchange of business documents and messages between information systems and services in the enterprise. The meta-data include information such as timestamp, sender and receiver of documents or messages. Events may be recorded by various logging systems, e.g., workflow logs, Web services logs, Web logs, and email logs (See Figure 5.1). Examples of such systems are included in SAP Netweaver, IBM Websphere, and HP SOA Manager. The level of details of information recorded in logs varies with logging systems.
- **Business data sources** which refer to data sources that contain the business documents and messages that are exchanged or task execution data that are produced during business process executions. This data may be stored as files of various formats in file systems and repositories, tuples in the database, or as text in Email systems (see Figure 5.1 for some examples).

These two types of data sources are maintained separately, or in some cases using the same systems. For instance, B2B hub may record meta-data about the exchange of documents, but not the actual documents that are exchanged, and the actual documents may be stored in a document management system. In other cases, the event related to the exchange of an XML message between Web services along with the message may be stored in the same service log.

The superimposition of a process metaphor over the data sources requires: (i) if the events data sources and business data sources are separate, identification of the *correspondence* business documents and event data. This is because neither of this information may be sufficient by itself to allow interpretation of data in terms of process execution. An *information item* refers to an event and its correspondent business data. (ii) *mappings* of information items to the progression of process tasks (start, execution and completion of process tasks). A *process item* refers to an information item mapped to a process tasks. (iii) definition of criteria or mechanisms to *correlate* process items into process instances. A process instance represent the tasks performed during a process execution to achieve a business goal, and (iv) the *model* of the underlying process followed by the process instances. This can be used as a reference for asking queries.

Therefore, the logical components of a data space are: data sources, correspondences, mappings, correlations, and process models. As mentioned in the introduction, different mappings, correlations and process models can be defined on the same set of information items. This is not only because the underlying information may belong to different processes, but also because different users (analysts) look at the same data from different perspectives. For instance, consider information related to purchase orders $PO1$, $PO2$, $PO3$, $PO4$, $PO5$ and $PO6$, which belong to 6 different process instances from the perspective of a Purchase Order manager. However, if $PO1$ and $PO2$ are shipped together, and $PO3$, $PO4$, $PO5$ and $PO6$ together, then they belong to only two different instances from the Warehouse Manager's perspective. A *process view* refers to a given way of mapping, correlation, and corresponding process model. Figure 5.2 shows process views defined in a pro-

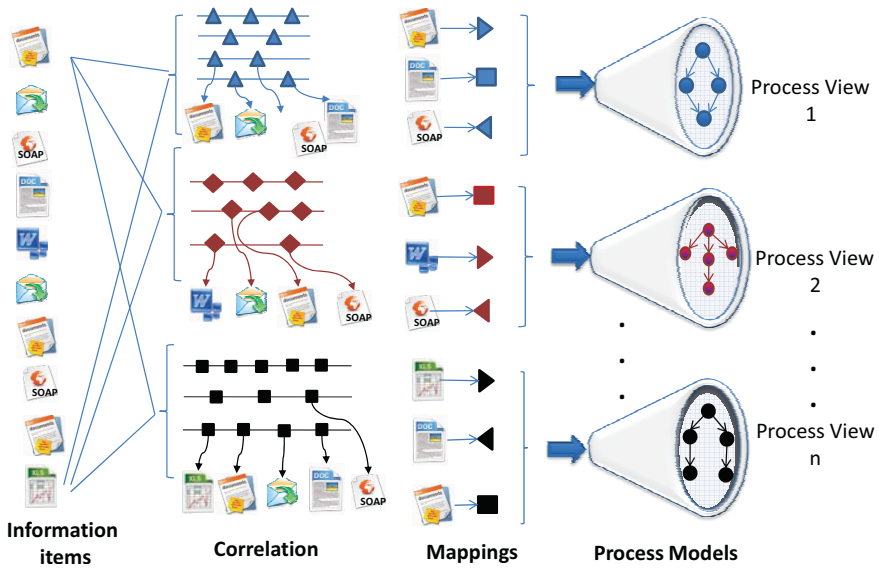

Fig. 5.2 Process views defined over information items characterize a process space

cess space starting from the information items. In other words, It is also possible to characterize a process space using process views defined over information items.

A process view may be nested. For instance, process view of the purchase order management system is nested within that of the whole enterprise, considered as a sub-process. This allows to look at a process space at various levels of abstractions and granularities from the high-level (enterprise-level) to details of a process execution. This covers the needs of various users in the enterprise.

5.3.3 Process Space Management System

Process space management system (PSMS) [350] proposed to enable interpretation of information in the process space. A PSMS offers the following "typical" categories of functionalities: *process space definition/discovery*, *process space analysis*, and *end user tools for process space exploration and visualization*. Figure 5.3 shows an example of an architecture for a PSMS organized in three layers on top of data resources in a process space.

Process space definition/discovery. The main step towards development of a PSMS is to define the process space that is to identify the logical components of the process space. This can be done both by human users to manually define, or automatically discovered from the data sources in the process space. Therefore,

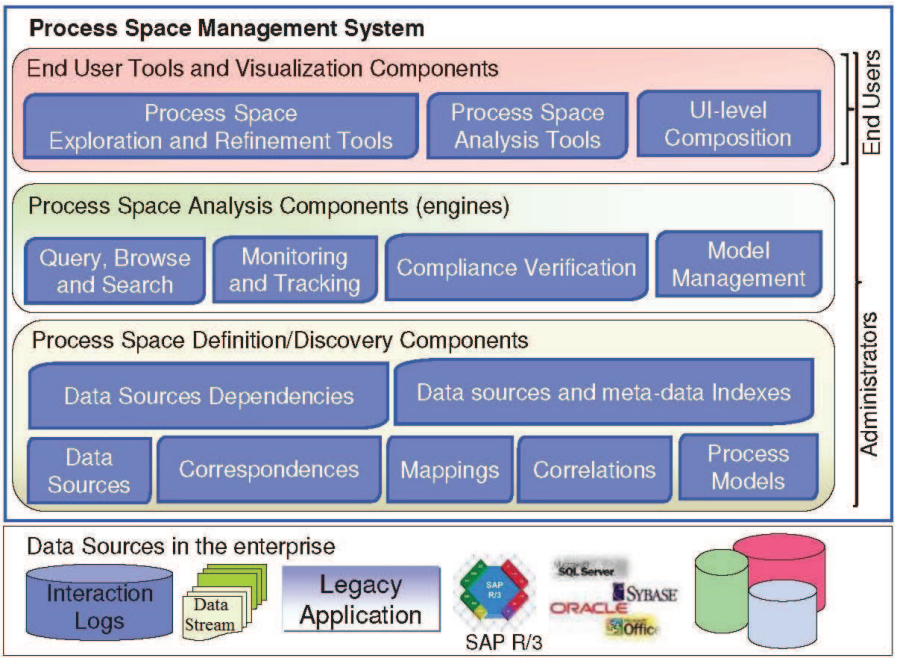

Fig. 5.3 The components of a process space management system

the PSMS should allow users to manually define, update and maintain information about these components. It also should provide automated support for discovery of the components. In particular, it must support incremental discovery and evolution of the process space components as more information become available in the enterprise.

In order to facilitate this task, the system should maintain a catalogue of the metadata about each component of the process space. For instance, for a data source this information include its name, location, owner and the names of processes that their information can be found in the data source. Similarly, the catalogue may contain information about the correspondences including evidences that the events and business data are related. Other information in the catalogue includes a description of criteria used for correlation of process items, and also mappings. In addition, the method that information is created, its creator and the degree of confidence in the information, if they are discovered automatically, have to be stored.

In addition to the logical components of the process space, there are two other first class objects in the process space, that are used in many other higher layer components, which are described in the following.

Indexes. Indexes can also be considered as first class objects in the process space. Indexes may be created on all process-related information on data sources, and more

importantly on the meta-data stored about the process space logical components. The reason is that indexes support performing efficient searching, querying and analysis of the process space. Indexes can be seen as summary information about the process space that can be used to answer many queries without the need to (re-)compute them from the process space. This component should be adaptive, i.e., allow to update the indexes as the data sources become (un)available. The indexing mechanism may allow accessing the same process item via multiple references to it, e.g., by process instance, by process name, by the method of correlation, etc.

Dependencies. Another first class object in the process space is the relationship (dependencies) among information systems and services in terms of process execution. This information can identify the role of each information system in the process, and also on systems that it depends to function. This information also allows to have a profile of information systems to know e.g., which systems cooperate in processing an order, and which services can be impacted by the failure of a service or system.

Process space analysis components. As depicted in Figure 5.3, a PSMS offers several interrelated components for analyzing, querying, and monitoring of process executions in process spaces, some of which are generations of components provided by traditional WfMS. However, unlike a WfMS which owns and controls all the process-related data, a PSMS allows the information to be independently managed by various information systems, and it provides a new set of services over the aggregate of these systems. In particular, the following analysis tasks are desired to be supported: (i) Browse, query and search engine; (ii) Monitoring and tracking engine; (iii) Compliance verification engine; (iv) Compliance verification engine; (vi) Model management engine; and (vii)Process space exploration and visualization tools.

Performing OLAP-style analysis on executions is also another type of queries. For instance, queries such as *how many people on average touch an order? what is the overhead and delays caused by human interaction vs machines? Which factors contribute to a process execution falling below the desired quality targets?*; In addition, the system should be able to search the content of documents and event logs for process-related information, e.g., based on keywords. Examples of such search queries include *find all purchase orders in which CPUs are requested*. As process space is potentially huge, it is important that the system can help users in formulating queries, e.g., by providing hints, identifying (non-)plausible queries, and proposing visualized approaches for query formulation.

Browse, query and search engine. This component helps in browsing, querying, and searching the process space. The information in the catalogue can be used to browse different information systems and repositories (e.g., logs and document management systems) that contain information on process executions, and understand the correspondences, mappings and correlations between them. Another functionality of this component is enabling querying of logs and documents. This includes queries that uses data sources from different systems (e.g., SOAP messages logs and emails related to the processing of the same order). An example of such queries is *select all orders approved by John where the time elapsed between or-*

der receipt and order shipment is more than 20 *days*. Note that not all data sources may support the query of the same level of expressiveness, and also they may not support queries with process information. In this case, a PSMS may extend these data sources to make them process-aware or to translate the queries into appropriate queries for such data sources.

Performing OLAP-style analysis on executions is also another type of queries. For instance, queries such as *how many people on average touch an order? what is the overhead and delays caused by human interaction vs machines? Which factors contribute to a process execution falling below the desired quality targets?*; In addition, the system should be able to search the content of documents and event logs for process-related information, e.g., based on keywords. Examples of such search queries include *find all purchase orders in which CPUs are requested*. As process space is potentially huge, it is important that the system can help users in formulating queries, e.g., by providing hints, identifying (non-)plausible queries, and proposing visualized approaches for query formulation.

The searching and querying functionality should allow posing queries at various levels of abstractions (e.g., at the enterprise level or at the process execution level). The search on the meta-data about the process space, e.g., on the data sources, correspondences, mappings, correlations and even process models are other types of required querying and search.

Monitoring and tracking engine. Monitoring of process execution and tracking the progress of a given instance is one of the important functionalities required by users in the enterprise. Examples of questions that could be answered by monitoring include *which route did a given order take? where did it get stuck? who has been working on it? where was the bottleneck? has it been shipped? has it been paid?* To enable answering above questions, the system enables accessing to up to date information about the process execution. The monitoring may also be performed at various levels of abstractions.

Compliance verification engine. This component enables to verify compliance of process executions in the process space with policies and regulations. For instance we may be interested to know if *privacy policies respected by the way information are processed in the process space?*

Model management engine. This component provides operators for analysis and management of process models at various levels of abstraction. Example of operators include identifying *subsumption*, *replaceability*, *compatibility* and *part-of* relationships between process models. These are helpful to understand if the model followed in the process space complies to the one designed in the enterprise, and also the relationship between processes executed by different systems, and also if there is any overlap between them.

End user tools and visualization components. Providing tools and visualization techniques to explore the data space and perform analysis tasks are required for end users. In particular, the following supports are useful in PSMS:

Process space exploration and visualization tools. These include tools that provide visual assistance to process space administrators in discovery process space components and other first class objects and also refining and maintenance of meta-

data about these components. This may be provided by integrating or extending existing successful visualization and user interface paradigm such as spreadsheets to perform light process-related analysis tasks for end users. Spreadsheets are widely used end user tools due to their simplicity, flexibility and relative richness of analysis. Many vendors have already integrated spreadsheets into workflow systems such as OracleBI [371] and SAP BI [414]. The intention here is to take this to a higher level by making spreadsheet-like environments available for business process analysis and management in process spaces for end users.

Process analysis tools. These component provide user interface for performing the analysis operations offered by the components in the lower level. In particular, the provided interfaces may offer graphical environments to perform business process intelligence tasks such as process monitoring, performance evaluation and measurements, e.g., in terms of KPI measures, process verification, and offer model management tools to use the engines over the process space.

User interface level composition. Once users have known the resources available in their process space, in terms of process execution, they may be able to build new applications by reusing existing applications. End users would appreciate tools that allow them to perform composition at the user interface level. Hence, this component should allow users to compose the functionality of various information systems in the process space and build mashups, e.g., using visual components such as pipes and widgets.

Using the current technology to achieve the functionalities that should be offered by a process space management system requires extensive (process-specific) development effort. This may be require performing ad hoc, tedious and error prone tasks as discussed briefly earlier. First step in building a process space is to identify its logical components. Identification can be achieved through manual definition of these components in the system and also by discovering information about them through automated approaches. Querying, browsing and searching process spaces require having appropriate process analysis languages. Using existing technologies, we would have to resort to SQL queries and to ad hoc programs to extract the information we want from the data. However, the abstraction level of the information we want to extract is different from that of the analysis language (SQL). What is needed here is a way to ask high level "process queries" or to perform process browsing, possibly with the same level of simplicity that IT managers are used when dealing with reporting on Excel, or in Web search. Visual query languages for business processes have been proposed [55].

In traditional data integration approaches (e.g., for building data warehouses or federated databases), data cleaning (consisting of detecting and correcting errors and inconsistencies from data) is an important step to improve the quality of data. Data of poor quality might cause significant cost escalation and time delays on business processing that relies on the data, e.g., mailing cost increase (posting mails to wrong addresses) and delivery delays (due to wrong addresses). Like data warehousing or decision support systems, the problem of cleaning process data (including events) is also one of major problems in the area of process spaces. Data quality in process spaces is affected by the following reasons: (i) as in traditional data integration, data

entry errors, inconsistent conventions, and poor integrity constraints, and (ii) bugs and flaws in the implementation of logging systems or the business process, exceptions in the process execution, and abnormal termination of the process interactions. Such data imperfection could lead to incorrect results in analysis results, erroneous process space discovery, inaccurate data correlation, etc. Hence, it is required to develop a data cleaning approach that effectively identifies the characteristics of process execution data and resolves the differences or conflicts of the data.

A process space must cover a wide range of heterogeneous data sources and applications. One major issue to accessing such systems is the inherent uncertainty about the correctness of information and the availability of some desired services over all data sources due to issues in the integration of heterogeneous data sources such as the level of supported interoperation, the expressiveness of supported query languages, noise, and various performance (e.g., legacy systems) that may affect the response time of the system. So a PSMS may be able to provide its service with variable level of guarantees on different data sources. Indeed, achieving the same level of guarantees that a WfMS offers may not be possible. The research challenge is then how to define realizable, practical and meaningful levels of service guarantees, and defining existing trade-offs and factors which influence the quality and performance measures. Another related research challenge is how to make a PSMS robust, i.e., tolerant to the inaccuracies in the data sources and to follow the "best-effort" model in returning results of analysis.

5.4 Process Mining

In order to analyze process execution data, querying execution logs of completed business processes (i.e., process mining [5, 10]) received continuous attention in research. The goal of process mining is to simplify process queries and to semi-automate the query formation in order to easily establish links between the actual processes, their data, and the process models. Process mining subsumes process analytics and enhances it with more comprehensive insights on the process execution: process mining techniques can be used to identify bottlenecks and critical points through *replaying* the execution traces used to discover a process model and *enrich* the discovered model with quantitative information.

In particular, process mining helps in discovering and improving real processes by extracting knowledge from event logs through using process modeling/analysis, machine learning, and data mining techniques. The main concern of these approaches is to reverse engineer the definitions of business process models from execution logs of information system components. Moreover, depending on how much details the log gives, they can provide statistics about many aspects of the business processes such as: the average duration of process instances or the average resource consumptions.

Recently, the IEEE Task Force on Process Mining released a manifesto describing guiding principles and challenges in process mining [8], where the goal is to

increase the maturity of process mining as a new tool to improve the (re)design, control, and support of operational business processes. In particular, process mining challenges include [5, 8, 10, 12]:

- mining hidden and duplicate tasks: one of the basic assumptions of process mining is that each event is registered in the log. Consequently, it is challenging to find information about tasks that are not recorded. Moreover, the presence of duplicate tasks is related to hidden tasks and refers to the situation that one can have a process model with two nodes referring to the same task.
- loops: in a process it may be possible to execute the same task multiple times, i.e., this typically refers to a loop in the corresponding process model.
- temporal properties: the temporal metadata (e.g., event timestamps) can be used for adding time information to the process model or to improve the quality of the discovered process model.
- mining different perspectives: understanding process logs in terms of its scope and details is challenging specially as it is subjective: depend on the perspective of the process analyst.
- dealing with noise and incompleteness: the log may contain noise (e.g., incorrectly logged information) and can be incomplete (e.g., the log does not contain sufficient information to derive the process).
- gathering data from heterogeneous sources: in modern enterprises, information about process execution is scattered across several systems and data sources.
- visualization techniques: helps presenting the results of process mining in a way that people actually gain insight in the process.
- delta analysis: is used to compare the two process models and explain the differences. It can be useful as process models can be descriptive or normative.

As illustrated in Figure 5.4, tree types of process mining are recognized: (i) discovery: this technique takes an event log and produce a model without using any a priority information. For example, the α-algorithm [13] takes an event log and produce a Petri net [358] model which explains the behavior recorded in the log; (ii) conformance: in this technique an existing process model is compared with an event log of the same process. Conformance checking can be used to check if reality, as recorded in the log, conforms to the model and vice versa. For example, the conformance checking algorithm proposed in [406] can be used to quantify and diagnose deviations; and (iii) enhancement: this technique can be used to extend or improve an existing process model using information about the actual process recorded in some event log. In this context, two types of enhancement are recognized: repair, can be used to modify the model to better reflect reality, and extension, can be used to add a new perspective to the process model by cross-correlating it with the log.

A recent book [5] and surveys [6, 10, 12] discuss the entire process mining spectrum from process discovery to operational support. Moreover, growing number of software vendors added process mining functionality to their tools. For example, ProM [9] offers a wide range of tools related to process mining and process analysis. In particular, ProM is a workflow discovery prototype tool that offers some of

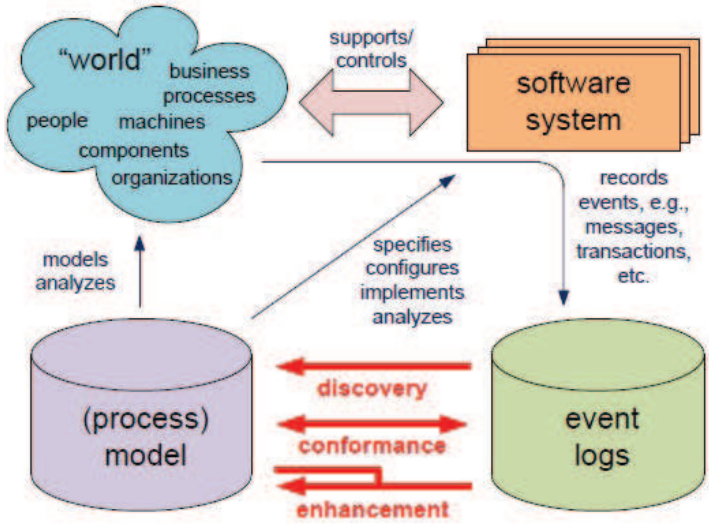

Fig. 5.4 Positioning of the three main types of process mining: (a) discovery, (b) conformance checking, and (c) enhancement [5].

above approaches. Agrawal et al. [24] proposed an approach to apply process mining in the context of workflow management systems and to address the problem of model construction. Datta [131] proposed algorithms for the discovery of business process models. Also, similar approaches in the the context of software engineering processes have been addressed in [126]. Herbst [231] presented a learning algorithm that is capable of inducing concurrent workflow models. The proposed approach focused on processes containing duplicate tasks and presented a specialization-based technique for discovering sequential model of process logs represented using HMM (Hidden Markov Model).

Conformance checking techniques [20, 357] are used to relate events in the log to activities in the model. Adriansyah et al. [20] presented techniques to measure the conformance of an event log for a given process model. The approach quantifies conformance and provides intuitive diagnostics and has been implemented in the ProM framework. Munoz-Gama et al. [357] presented an approach to enrich the process conformance analysis for the precision dimension. Some other examples of approaches focused on precision for: measuring the percentage of potential traces in the process model that are in the log [198], comparing two models and a log to see how much of the first models behavior is covered by the second [330], comparing the behavioral similarity of two models without a log [153], and using minimal description length to evaluate the quality of the model [99].

Enhancement techniques heavily rely on the relationship between elements in the model and events in the log. These relationships may be used to: (i) replay the event log on the model, e.g., bottlenecks can be identified by replaying an event

log on a process model while examining the timestamps [5]; (ii) modify the model to better reflect reality; and (iii) add a new perspective to the process model by cross-correlating it with the log. Subramanian et al. [433] proposed an approach for enhancing BPEL engines with facilities that permit satisfying self-healing requirements. Moreover, the concept of self-healing as a part of autonomic computing has been proposed in [265], where self-healing systems will automatically detect, diagnose, and repair localized problems resulting from failures. A diagnostic reasoning techniques and diagnosis-aware exception handlers for exception handling proposed in [38]. Also, a framework for providing a proxy-based solution to BPEL, as an approach for dynamic adaptation of composite Web services, presented in [169].

Process Mining vs. Data Mining. Although process mining and data mining have lots of aspects in common, there are also fundamental differences in what they do and where they can be useful. The goal of data mining is to discover previously unknown interesting patterns in datasets. In this context, various methods at the intersection of database techniques such as spatial indices, artificial intelligence, machine learning, and statistics can be used. Data mining provides valuable insights through analysis of data and does not have concerns about the processes. This is where process mining provides the opportunity to get the same benefits of data mining, when working with processes and focusing on process improvements. In this context, unlike data mining, process mining focuses on the process perspective to find process relationships in the data. More specifically, process mining's perspective is not on patterns in the data but in the processes the data represents. Process mining can be seen as the 'missing link' between data mining and traditional Business Process Management. From the similarity point of view, both process mining and data mining use the mining techniques to analyze large volumes of data, e.g. process logs in process mining and electronic health records (EHR) in data mining. Moreover, both techniques produce information that can be helpful for making business decisions.

5.5 Analyzing Cross-cutting Aspects in Processes' Data

Modern business processes have flexible underlying process definition where the control flow between activities cannot be modeled in advance but simply occurs during run time [154]. The semistructured nature of such process's data requires analyzing process related entities, such as people and artifacts, and also the relationships among them. In many cases, however, process artifacts evolve over time, as they pass through the business's operations. Consequently, identifying the interactions among people and artifacts over time becomes challenging and requires analyzing the cross-cutting aspects [161] of process artifacts. In particular, process artifacts, like code, has cross-cutting aspects such as versioning (what are the various versions of an artifact, during its lifecycle, and how they are related) and provenance [116] (what manipulations were performed on the artifact to get it to this point).

The specific notion of business artifact was first introduced in [364] and was further studied, from both practical and theoretical perspectives [83, 185, 238]. However, in a dynamic world, as business artifacts changes over time, it is important to be able to get an artifact (and its provenance) at a certain point in time. It is challenging as annotations assigned to an artifact (or its versions) today may no longer be relevant to the future representation of that artifact: artifacts are very likely to have different states over time and the temporal annotations may or may not apply to these evolving states. Consequently, analyzing evolving aspects of artifacts (i.e. versioning and provenance) over time is important and will expose many hidden information among entities in process data. This information can be used to detect the actual processing behavior and therefore, to improve the ad-hoc processes.

As an example, knowledge-intensive processes, e.g., those in domains such as healthcare and governance, involve human judgements in the selection of activities that are performed. Activities of knowledge workers in knowledge intensive processes involve directly working on and manipulating artifacts to the extent that these activities can be considered as artifact-centric activities. Such processes, almost always involves the collection and presentation of a diverse set of artifacts, where artifacts are developed and changed gradually over a long period of time. Case management [437], also known as case handling, is a common approach to support knowledge-intensive processes. In order to represent cross-cutting aspects in processes, there is a need to collect meta-data about entities (e.g., artifacts, activities on top of artifacts, and related actors) and relationship among them from various systems/departments over time, where there is no central system to capture such activities at different systems/departments.

Many approaches [81, 122, 185, 238] used business artifacts that combine data and process in a holistic manner and as the basic building block. Some of these works [185, 238] used a variant of finite state machines to specify lifecycles. New line of works, such as [58, 155], consider an artifact-centric activity model for business processes. These models support timed queries and enables weaving cross-cutting aspects, e.g., versioning and provenance, around business artifacts to imbues the artifacts with additional semantics that must be observed in constraint and querying ad-hoc processes.

5.6 Provenance and Evolution of Business Artifacts

Provenance refers to the documented history of an object (e.g. documents, data, and resources) or the documentation of processes in an object's lifecycle, which tracks the steps by which the object was derived [116] and evolved. This documentation (often represented as graphs) should include all the information necessary to reproduce a certain piece of data or the process that led to that data [348]. The ability to analyze provenance data is important as it offers the means to verify business data products, to infer their quality, and to decide whether they can be trusted [347]. In a dynamic world, as data changes, it is important to be able to get a piece of data

as it was, and its provenance data, at a certain point in time. Under this perspective, the provenance queries may provide different results for queries looking at different points in time. In this context, enabling time-aware querying of provenance information is challenging and requires explicitly representing the time information in the provenance graphs, and also providing abstractions and efficient mechanisms for time-aware querying of provenance graphs over an ever increasing volume of data.

Prior work on modeling and representing provenance metadata [116, 175, 423] (e.g. lineage, where-provenance, why-provenance, dependency-provenance, how-provenance, and provenance-traces models) model provenance as a directed acyclic graph, where the focus is on modeling the process that led to a piece of data. They present vocabularies to model process activities and their causal dependency, i.e. the relationship between an activity (the cause) and a second activity (the effect) where the second activity is understood as a consequence of the first. In a dynamic world data changes, so the graphs representing data provenance evolve over time. It is important to be able to reproduce a piece of data or the process that led to that data for a specific point in time. This requires modeling *time* as a first class citizen in the provenance models. Times, intervals, and versioning can be very important in understanding provenance graphs as the structure of such graphs evolves over time. Today's approaches in modeling provenance, e.g. in OPM, treat time as a second class citizen. Considering time as a first class citizen, will enable retrieving multiple snapshots of entities (versions) in the past which can help in capturing the provenance for each version of an entity independently. Moreover, it can help in understanding the role of each entity in the temporal context of the entire system.

To understand the provenance, consider a scenario based on breast cancer treatment cases in Velindre hospital [437]. Figure 5.5-A represents a case instance, in this scenario, where a General Practitioner (GP) suspecting a patient has cancer, updates patient history, and referring the patient to a Breast Cancer Clinic (BCC). BCC checks the patients history and requests assessments such as an examination, imaging, fine needle aspiration, and core biopsy. Therefore, BCC administrator refers patient to Breast Cancer Specialist Clinic (BCSC), Radiology Clinic (RC), and Pathology Clinic (PC), where these departments apply medical examinations and send the results to Multi-Disciplinary Team (MDT). The results are gathered by the MDT coordinator and discussed at the MDT team meeting involving a surgeon oncologist, radiologist, pathologist, clinical and medical oncologist, and a nurse.

Analyzing the results and the patient history, MDT will decide for next steps, e.g., in case of positive findings, non-surgical (Radiotherapy, Chemotherapy, Endocrine therapy, Biological therapy, or Bisphosphonates) and/or surgical options will be considered. During interaction among different systems, organizations and care team professionals, a set of artifacts will be generated. Figure 5.5-B represents parent artifacts, i.e., ancestors, for patient history document, and Figure 5.5-C represents parent artifacts for its versions. Figure 5.5-D represents a set of activities which shows how version v_2 of patient history document develops and changes gradually over time and evolves into version v_3.

Considering this scenario, modeling and analyzing provenance data will enable process analyst to answer the following questions: Who was involved in generating

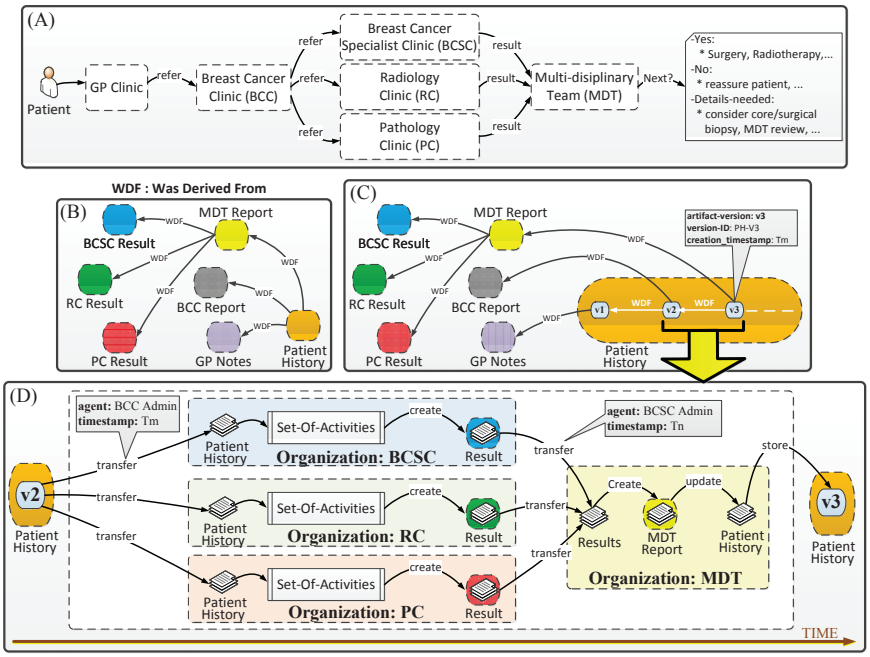

Fig. 5.5 Example case scenario for breast cancer treatment including a case instance (A), parent artifacts, i.e. ancestors, for patient history document (B) and its versions (C), and set of activities which shows how version v_2 of patient history document develops and changes gradually over time and evolves into version v_3 (D).

an artifact? What are the changes applied to the artifact over different points of time? and who was involved in these processes? How one version of the artifact evolved from another version? What was the used artifact and the purpose to obtain result? What was the used artifact and the collected data used to obtain result? What was the used artifacts and the security processes applied to it? etc. Several provenance models [116, 175, 423] have been presented in a number of domains (e.g. databases, scientific workflows and the Semantic Web), motivated by notions such as influence, dependence, and causality. Why-provenance [175, 423] models the influences that a source data had on the existence of the data. Where-provenance [116] focuses on the dependency to the location(s) in the source data from which the data was extracted. How-provenance [116, 175] represents the causality of an action or series of actions performed on or caused by source data. Discovering historical paths through provenance graphs forms the basis of many provenance query languages-queries [233, 260]. Temporal databases [344] enable retrieving multiple snapshots (versions) of data artifacts at different points in time. However, a temporal database does not capture important information for data provenance such as activities performed on the data, agents acting on the data, and the relationships that the different versions of artifacts have to each other in various points in time. Approaches for

modeling and querying graphs (e.g. [36, 60, 194]) can be used for querying provenance data.

Besides provenance and versioning, other aspects of business artifacts such as security (who has access to the artifact over time), privacy (what actions were performed to protect or release artifact information over time), and trust (the credibility of users and their posted and shared content in a particular domain) need to be analyzed. Analyzing these aspects will expose many hidden interactions among entities in process graphs. For example, in current outsourcing practices, clients usually focus primarily on business objectives and security is negotiated only for communication links. In such scenarios, strong protection of a communication link is of little value if data can be easily stolen or corrupted while on a supplier's server. For example, analyzing such manipulated data (e.g. stolen or unauthorized accessed data), may lead to unreliable decisions.

http://www.springer.com/978-3-319-25036-6

Process Analytics
Concepts and Techniques for Querying and Analyzing Process Data

Beheshti, S.-M.-R.; Benatallah, B.; Sakr, S.; Grigori, D.; Motahari-Nezhad, H.R.; Barukh, M.C.; Gater, A.; Ryu, S.H.

2016, XVI, 178 p. 30 illus., 6 illus. in color., Hardcover
ISBN: 978-3-319-25036-6