Declarative Process Mining

When Event-Based Systems (EBSs) are executed inside a company, it is usually the case that all relevant event occurrences are collected and stored inside an information system. All the modern Business Process Management (BPM) systems used in the industry provide logging facilities.

The importance of logging the performed operations is twofold. On the one hand, audit legislation (such as the Sarbanes-Oxley Act) needs complete information about past execution traces (also called audit trails), to determine whether the boundaries imposed by external regulations and norms are respected \cite{250}; violation of rules enforced by law indicates malpractices or frauds. On the other hand, the extraction of information from event logs is of utmost importance also for the company itself. Indeed, while the designed specifications are strongly influenced by the perceptions of the modeler, event logs reflect reality, and can help business analysts and managers in the formation of an objective view of their organization.

The extraction of information from event logs is called process mining\footnote{http://www.processmining.org}. Process mining can be considered as a special case of data mining, where the emphasis is on data characterizing the dynamics of EBSs. Differently from run-time verification and monitoring, process mining techniques are applied a-posteriori, starting from complete traces that refer to past instances.

During the last years, process mining has started to drawn the attention of industry \cite{247}. Companies are interested in the application of process mining techniques on their own logs to point out inefficiencies, risky points, bottle-necks and sources of noncompliance. This information is then used to drive the revision and improvement of Business Processes (BPs), covering the diagnosis phase of the BP life cycle, shown in Fig. 2.1 – p. 21. Furthermore, process mining is a viable solution, because it is minimally intrusive. It simply requires to extract the event logs from the company’s information system, without any impact on its IT infrastructure.

\footnote{http://www.processmining.org}
Fig. 15.1 depicts three relevant process mining techniques [251]:

**Log-based verification** checks whether the recorded instances comply with some property or regulatory model. It is also called *trace analysis*.

**Conformance checking** compares traces with a pre-existing model, verifying whether the executed instances have effectively followed its prescriptions; it represents the “a-posteriori” version of the (a-priori) compliance verification task (see Sect. 12.1.2) – instead of statically analyzing an interaction model, it focuses on the real behavior of interacting entities.

**Discovery** aims at extracting a new interaction model starting from a set of execution traces; the obtained model accounts for reality, and can be therefore compared with pre-existing designed models to assess their discrepancy (*delta analysis*) as well as used to drive their revision and improvement.

In this chapter, we focus on *declarative process mining*, i.e., on techniques which rely on the declarative paradigm for extracting information from event logs. In particular, ConDec++ (Chaps. 3 and 6) and CLIMB/SCIFF (Chap. 4), together with corresponding reasoning techniques, are exploited to realize two process mining tools. Both tools have been implemented as *ProM* [238] plug-ins, and are part of its last distribution. ProM is one of the most widespread process mining frameworks.

We describe the peculiar features of the two plug-ins, reporting how they have been applied in the context of some real case studies.

---

2 Although process mining techniques have been initially applied to BPs and workflows, they can be seamlessly exploited for any EBS. Following the terminology proposed in this book, in its more general sense process mining then deals with “interaction models”.