Modeling and Monitoring Business Process Execution

Piergiorgio Bertoli[1], Mauro Dragoni[2], Chiara Ghidini[2], Emanuele Martufi[1], Michele Nori[1], Marco Pistore[1,2], and Chiara Di Francescomarino[2]

[1] SAYservice, Trento, Italy
{bertoli,martufi,nori,pistore}@sayservice.it
[2] FBK—IRST, Trento, Italy*
{dragoni,ghidini,pistore,dfmchiara}@fbk.eu

Abstract. The growing adoption of IT systems to support business activities has made available huge amount of data, that can be used to monitor the actual execution of business processes. However, in many real settings, due to the different degrees of abstraction between business and technological layers and to information hiding, the potentiality of this data cannot be fully exploited. The PROMO tool, grounded on reasoning services, aims at reconciling the technical and the business layer, in order to enable the effective monitoring and analysis of business process instances in the face of abovementioned issues.

1 Introduction

Nowadays, huge quantities of data are made available by the growing capability of Information Technology (IT) systems to trace and store business service and application execution information. The potentiality of this data is enormous from a business point of view, as it makes it possible (i) to observe the current evolution of ongoing processes; (ii) to provide statistical analysis on past executions; (iii) to detect deviations of real process executions from ideal process models (as envisaged in [1]); (iv) to identify performance-specific or instance-specific problems; and hence also to improve business process models based on analyses, deviations, bottlenecks and problems detected inspecting real process executions. Indeed, a variety of Business Intelligence tools have been proposed, even by major vendors, that aim at supporting business activity monitoring (BAM) to different extent; for instance, Engineering's eBAM [5], Microsoft's BAM suite in BizTalk [4], Oracle's BAM [6], Polymita, WebSphere [7], to name a few.

However, business activity monitoring must deal with a significant difficulty, i.e., the gap existing between the business and the technological (IT) level. Indeed a perfect mapping between modeled and IT-traced processes does not exist in the vast majority of cases. For example, observation of process execution often brings (e.g., because of manual activities or paper-based documentation) only partial information in terms of which process activities have been executed and what data or artifacts they have produced so far. Moreover, even when IT information exists, it is not easy to associate it to

* This work is supported by "ProMo - A Collaborative Agile Approach to Model and Monitor Service-Based Business Processes", funded by the Operational Programme "Fondo Europeo di Sviluppo Regionale (FESR) 2007-2013" of the Province of Trento, Italy.

S. Basu et al. (Eds.): ICSOC 2013, LNCS 8274, pp. 683–687, 2013.

a specific process instance execution. Indeed, IT-services can be shared among several process classes and instances, and the traced information can be hard to disambiguate. To the best of our knowledge, none of the aforementioned current approaches has completely tackled the above issues, rather relying on the strong simplifying assumption that the business analyst can link directly IT information to business process activities.

In this demo we present PROMO[1], a software tool which, exploiting reasoning services, aims at reconciling the technical layer and the business one to enable the effective monitoring, analysis and querying of business processes.

2 PROMO Approach and Tool

PROMO aims at providing a collaborative (involving *domain experts* and *IT experts*) approach to model, monitor and analyze business processes, filling the unavoidable gap existing between business and IT layers. To this purpose PROMO introduces an intermediate layer (see Figure 1), which enables the communication between the business and the technological one through an intermediate model. Such a model, able to formalize the relationships between the business models and the information extracted at IT level, relies on the integrated representation of all the information collected about the process execution (named *IT-trace*).

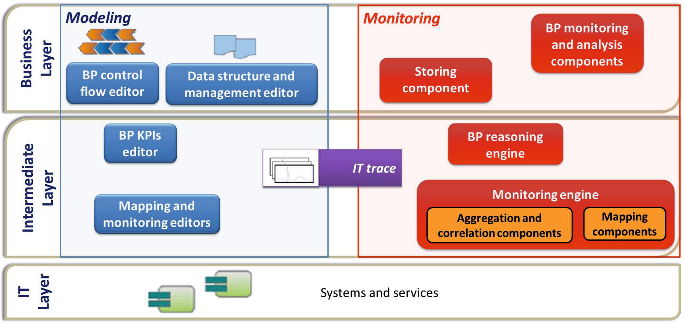


Fig. 1. PROMO overview

To accomplish its goal, PROMO integrates a modeling component and a monitoring component. At the business level, the **modeling component** provides MOKIPRO, a customized version of the MediaWiki-based[2] tool MOKI [3] for the process and ontology editing, which allows *domain experts* to design the business process control flow, as well as the associated data representation and manipulation (by the process activities). MOKIPRO (i) customizes the ORYX editor[3] for the BPMN modeling of business processes and related data structures (see Figure 2); (ii) allows *domain experts* to specify the data structure and, for each process activity, the data that it creates or shows.

[1] A video of the demo is downloadable from
https://dkmtools.fbk.eu/moki/icsocVideoDemo.zip

[2] http://www.mediawiki.org

[3] http://bpt.hpi.uni-potsdam.de/Oryx/

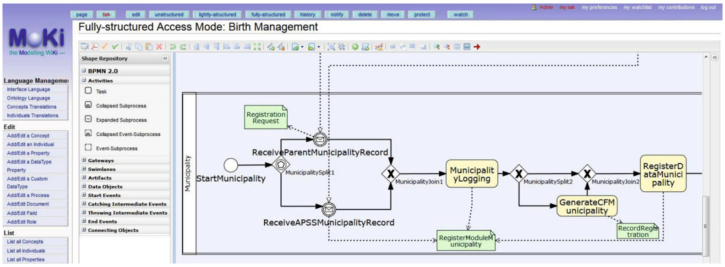
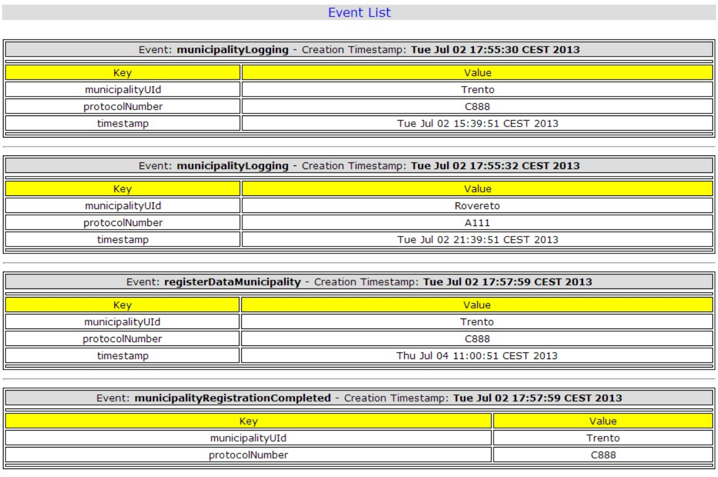


Fig. 2. Process modeling in PROMO

At the IT-level PROMO provides (i) mapping and monitoring editors that allow *IT experts* (taking advantage of the *domain experts* modeling) to specify, respectively, aggregation/monitoring rules and the relationships between business models and the information extracted at IT level; and (ii) a KPI-editor for the definition of interesting KPIs to be monitored. In detail, the DomainObject language [2] is used for defining mapping properties, an ad-hoc rule language for the monitoring rules, while business KPIs are defined as SPARQL [8] queries. It is possible for example to monitor how many times process instances deviate from the "prescribed" behavior or a specific branch of the model is executed, as well as the average time required for the process execution.



Event List

Event: **municipalityLogging** - Creation Timestamp: **Tue Jul 02 17:55:30 CEST 2013**

Key	Value
municipalityUId	Trento
protocolNumber	C888
timestamp	Tue Jul 02 15:39:51 CEST 2013

Event: **municipalityLogging** - Creation Timestamp: **Tue Jul 02 17:55:32 CEST 2013**

Key	Value
municipalityUId	Rovereto
protocolNumber	A111
timestamp	Tue Jul 02 21:39:51 CEST 2013

Event: **registerDataMunicipality** - Creation Timestamp: **Tue Jul 02 17:57:59 CEST 2013**

Key	Value
municipalityUId	Trento
protocolNumber	C888
timestamp	Thu Jul 04 11:00:51 CEST 2013

Event: **municipalityRegistrationCompleted** - Creation Timestamp: **Tue Jul 02 17:57:59 CEST 2013**

Key	Value
municipalityUId	Trento
protocolNumber	C888

Fig. 3. Monitoring IT events in PROMO

At run-time, whenever an IT-level event occurs, the **monitoring component** captures and takes care of it. In detail, the event is managed by the monitoring engine, which, based on the specification and rules defined at design-time, correlates and aggregates events, produces new control events, monitors and maps the events to the corresponding one(s) at the business layer and eventually produces the IT-trace. Figure 3 shows a list of events monitored (and produced) by the monitoring engine in a specific scenario, which will be used in the demo.

The information in the IT-trace, which in many cases is only partial with respect to a complete execution flow of a designed process model, is hence passed to a reasoning engine, which reconstructs missing information by applying model-driven satisfiability rules. The reconstructed IT-trace is finally recorded in a semantic-based store to be monitored at business level (according to the KPIs defined at designed time), or queried (according to queries defined by *domain experts*), by the BP monitoring and analysis component. For example, specific queries investigating the number of times in which a business activity is executed by a given actor instance, or in which the activity provides as output a data structure field with a given value, can be formulated.

3 PROMO Application

The PROMO tool has been applied, among other case studies, to an e-government one: the Italian *Birth Management* procedure, characterized by a process containing 4 pools, 18 activities, 21 gateways, and 13 data structures (including in total 75 distinct fields). Our demo will showcase PROMO on this case study.

At design time, *domain experts* and *IT experts* model the control flow (Figure 2 depicts the initial part of the Municipality pool), data structure and manipulation by activities, as well as mapping definition and monitoring rules for the case study.

As an example, consider when, at run-time, a *MunicipalityLogging* event is registered by the services at the IT layer, captured and analyzed by the monitoring engine. Then, when another event related to the same process instance (e.g., *RegisterDataMunicipality*) occurs (see Figure 3), the monitoring engine correlates and aggregates them, generating a new IT event (*MunicipalityRegistrationCompleted*), used for IT-level monitoring. Further, the (partial) IT-trace is generated and passed to the reasoning engine, which tries to reconstruct missing information according to the process model. For instance, the reasoning engine, aware of the process model control flow, can infer that the execution went through either the *ReceiveParentMunicipalityRecord* or the *ReceiveAPSSMunicipalityRecord* activity. Moreover, knowing the data associated with the two received events (the data structure associated to the *MunicipalityLogging* activity does not contain the *Fiscal Code CF*, while the one associated to the *RegisterMunicipalityData* activity does), as well as the activities in charge to create or show them, the reasoning engine can infer that the process execution actually went through the *GenerateCFMunicipality* activity. The reconstructed trace is finally recorded by the storing component and the related KPIs updated. Among the predefined KPIs of this case study, for instance, there is the number of times in which the *CF* field has been filled by the Municipality rather than by another actor of the procedure and the average time required to complete the whole birth practice.

References

1. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes, 1st edn. Springer Publishing Company, Incorporated (2011)
2. Bertoli, P., Kazhamiakin, R., Nori, M., Pistore, M.: Smart: Modeling and monitoring support for business process coordination in dynamic environments. In: Abramowicz, W., Domingue, J., Węcel, K. (eds.) BIS 2012 Workshops. LNBIP, vol. 127, pp. 243–254. Springer, Heidelberg (2012)

3. Ghidini, C., Rospocher, M., Serafini, L.: Conceptual Modeling in Wikis: a Reference Architecture and a Tool. In: eKNOW 2012, Valencia, Spain, pp. 128–135 (2012)
4. Biztalk team: biztalk, biztalk server, `https://www.microsoft.com/en-us/biztalk/default.aspx`
5. Eclipse team: ebam, extended business activity monitoring, `http://www.eclipse.org/ebam`
6. Oracle team: Oracle bam, oracle business activity monitoring (oracle bam), `http://www.oracle.com/technetwork/middleware/bam/overview/index.html`
7. Websphere team: Websphere software, ibm, `http://www-01.ibm.com/software/websphere/`
8. Prud'hommeaux, E., Seaborne, A.: SPARQL query language for RDF (2008), Latest version available as `http://www.w3.org/TR/rdf-sparql-query/`