Using Process Mining to Generate Accurate and Interactive Business Process Maps

W.M.P. van der Aalst

Eindhoven University of Technology
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands.
w.m.p.v.d.aalst@tue.nl

Abstract. The quality of today's digital maps is very high. This allows for new functionality as illustrated by modern car navigation systems (e.g., TomTom, Garmin, etc.), Google maps, Google Street View, Mashups using geo-tagging (e.g., Panoramio, HousingMaps, etc.), etc. People can seamlessly zoom in and out using the interactive maps in such systems. Moreover, all kinds of information can be projected on these interactive maps (e.g., traffic jams, four-bedroom apartments for sale, etc.). Process models can be seen as the "maps" describing the operational processes of organizations. Unfortunately, accurate and interactive process maps are typically missing when it comes to business process management. Either there are no good maps or the maps are static or outdated. Therefore, we propose to *automatically generate business process maps using process mining techniques*. By doing this, there is a close connection between these maps and the actual behavior recorded in event logs. This will allow for high-quality process models showing what really happened. Moreover, this will also allow for the *projection of dynamic information*, e.g., the "traffic jams" in business processes. In fact, the combination of accurate maps, historic information, and information about current process instances, allows for *prediction* and *recommendation*. For example, just like TomTom can predict the arrival time at a particular location, process mining techniques can be used to predict when a process instance will finish.

1 The Need for Accurate and Interactive Business Process Maps

Process models are vital for the design, analysis, and implementation of information systems. Their role is similar to the role of maps for navigation systems, mashups, etc. For example, people increasingly rely on the devices of TomTom and other vendors and find it useful to get directions to go from A to B, know the expected arrival time, learn about traffic jams on the planned route, and be able to view maps that can be customized in various ways (zoom-in/zoom-out, show fuel stations, speed limits, etc.). Maps do not only play an important role in car navigation, but are also crucial for all kinds of innovative information services. Figure 1 shows two examples combining cartographic information

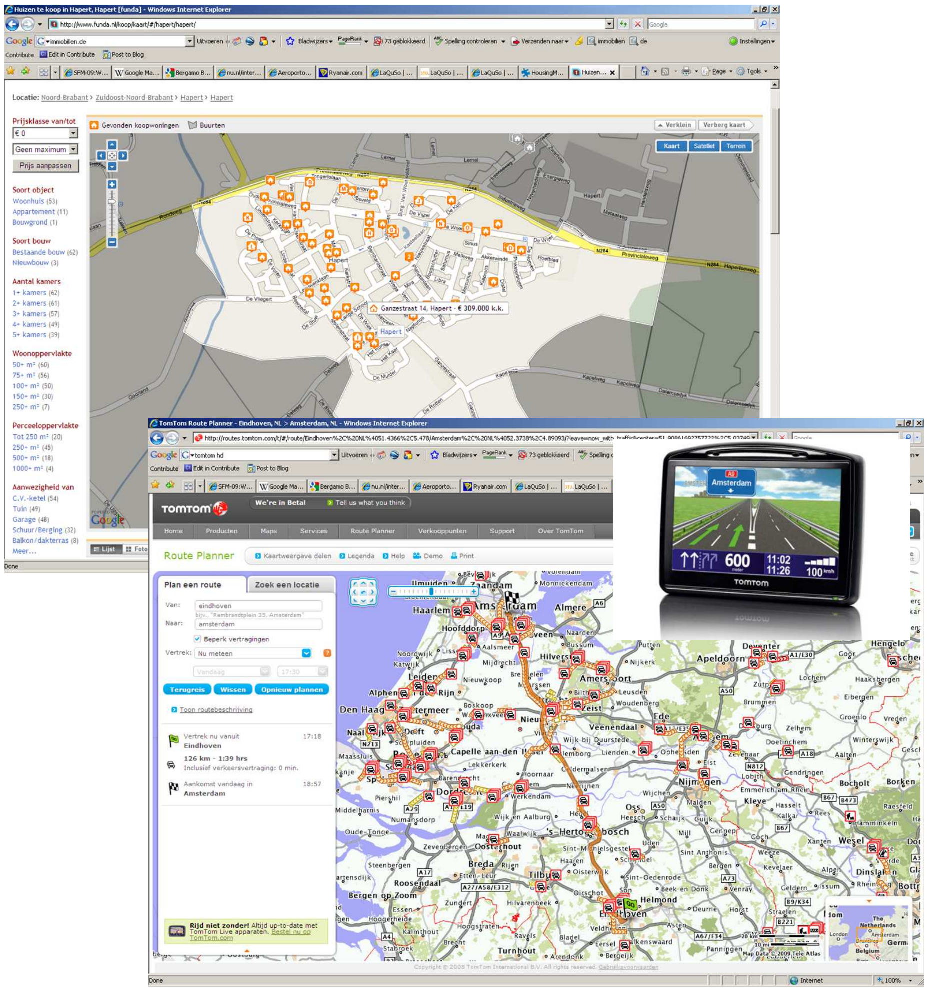

Fig. 1. The role of maps in Funda (top left) and TomTom HD Traffic (bottom right). Funda dynamically shows houses for sale in a particular area (in this case town of Hapert) meeting specific criteria (cf. www.funda.nl). TomTom HD Traffic is calculating the best route based on cell phone information provided by Vodafone, i.e., the locations and directions of cell phones are used to predict traffic jams (cf. www.tomtom.com). Both examples use a combination of high-quality maps augmented with dynamic information allowing for seamlessly zooming in and out. This paper advocates the development of such functionality for business information systems.

with dynamically changing data. However, when looking at business processes, such information is typically lacking. Good and accurate "maps" of business processes are often missing and, if they exist, they tend to be restrictive and provide little information. For example, very few information systems are able to predict *when* a case will complete. Therefore, we advocate more TomTom-like functionality for business process management, coined "TomTom4BPM" in [2]. Besides navigation systems, there are many applications based on Google maps. For example, real-estate agencies dynamically projecting information on maps, etc. A key element is the availability of high-quality maps. The early navigation systems were using very course maps that were often outdated, thus limiting their applicability. A similar situation can be seen when looking at information systems based on incorrect or outdated process models.

In this paper, we advocate the use of *accurate and interactive business process maps obtained through process mining*. The goal is to provide a better breed of *Business Process Management Systems* (BPMSs) [1, 15, 29]. BPMSs are used to manage and execute operational processes involving people, applications, and/or information sources on the basis of process models. These systems can be seen as the next generation of workflow technology offering more support for analysis. Despite significant advances in the last decade, the functionality of today's BPMSs leaves much to be desired. This becomes evident when comparing such systems with the latest car navigation systems of TomTom or innovative applications based on Google maps. Some examples of functionality provided by TomTom and/or Google maps that are generally missing in contemporary BPMSs are:

- In today's organizations often *a good process map is missing*. Process models are not present, incorrect, or outdated. Sometimes process models are used to directly configure the BPMS. However, in most situations there is not an explicit process model as the process is fragmented and hidden inside legacy code, the configuration of ERP systems, and in the minds of people.
- If process models exist in an explicit form, *their quality typically leaves much to be desired*. Especially when a process model is not used for enactment and is only used for documentation and communication, it tends to present a "PowerPoint reality". Road maps are typically of much higher quality and use intuitive colors and shapes of varying sizes, e.g., highways are emphasized by thick colorful lines and dirt roads are not shown or shown using thin dark lines. In process models, *all activities tend to have the same size and color and it is difficult to distinguish the main process flow from the less traveled process paths.*
- Most process modeling languages have a static decomposition mechanism (e.g., nested subprocesses). However, what is needed are controls allowing users *to zoom in or zoom out seamlessly like in a navigation system or Google maps*. Note that, while zooming out, insignificant things are either left out or dynamically clustered into aggregate shapes (e.g., streets and suburbs amalgamate into cities). Process models should not be static but allow for various (context dependent) views.

– Sometimes process models are used for enactment. However, such "process maps" are often trying to "control" the user. When using a car navigation system, the driver is always in control, i.e., the road map (or TomTom) is not trying to "control" the user. The goal of a BPMS should be to *provide directions and guidance rather than enforcing a particular route.*
– A navigation system continuously shows a clear *overview of the current situation* (i.e., location and speed). Moreover, traffic information is given, showing potential problems and delays. This information is typically missing in a BPMS. Even if the BPMS provides a management dashboard, TomTom-like features such as traffic information and current location are typically not shown in an intuitive manner.
– A TomTom system *continuously recalculates* the route, i.e., the recommended route is not fixed and changed based on the actions of the driver and contextual information (e.g. traffic jams). Moreover, at any point in time the navigation system is showing the *estimated arrival time*. Existing BPMSs are not showing this information and do not recalculate the optimal process based on new information.

The above list of examples illustrates desirable functionality that is currently missing in commercial BPMSs. Fortunately, recent breakthroughs in *process mining* may assist in realizing highly innovative features that are based on high-quality business process maps tightly connected to historic information collected in the form of event logs.

In the remainder of this paper, we first briefly introduce the concept process mining in Section 2. Section 3 introduces the ProM framework that aims at the generation of accurate and interactive business process maps obtained through process mining. Based on ProM and process mining it is possible to provide TomTom-like functionality as discussed in Section 4. One particular example of such innovative functionality is "case prediction" as described in Section 5. Pointers to related work on process mining are given in Section 6. Section 7 concludes the paper.

2 Process Mining

Process mining techniques attempt to extract non-trivial and useful information from *event logs* [5, 9]. Many of today's information systems are recording an abundance of events in such logs. Various process mining approaches make it possible to uncover information about the processes they support. Typically, these approaches assume that it is possible to sequentially record events such that each event refers to an activity (i.e., a well-defined step in the process) and is related to a particular case (i.e., a process instance). Furthermore, some mining techniques use additional information such as the performer or originator of the event (i.e., the person/resource executing or initiating the activity), the timestamp of the event, or data elements recorded with the event (e.g., the size of an order).

Process mining addresses the problem that most people have very limited information about what is actually happening in their organization. In practice, there is often a significant gap between what is prescribed or supposed to happen, and what *actually* happens. Only a concise assessment of the organizational reality, which process mining strives to deliver, can help in verifying process models, and ultimately be used in a process redesign effort or BPMS implementation.

Some examples of questions addressed by process mining:

- *Process discovery*: "What is really happening?"
- *Conformance checking*: "Do we do what was agreed upon?"
- *Performance analysis*: "Where are the bottlenecks?"
- *Process prediction*: "Will this case be late?"
- *Process improvement*: "How to redesign this process?"

The above questions show that process mining is not limited to control-flow discovery. In fact, we identify three types of process mining: (a) *discovery*, (b) *conformance*, and (c) *extension*. We also distinguish three different perspectives: (a) the *control-flow perspective* ("How?"), (b) the *organizational perspective* ("Who?") and (c) the *case perspective* ("What?").

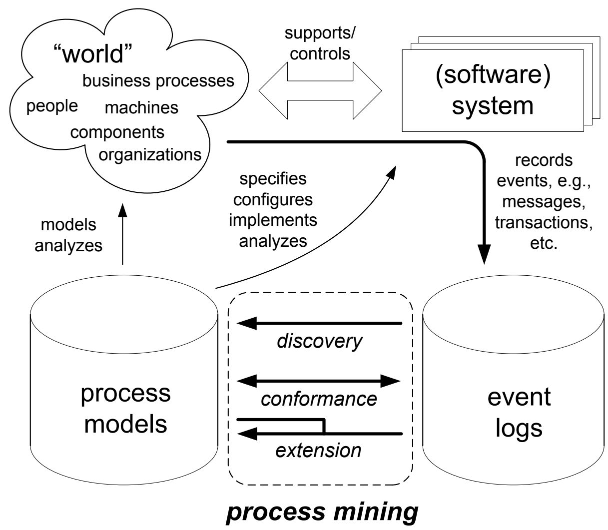

Fig. 2. Process mining as a bridge between process models and event logs.

Figure 2 positions process mining as the technology that "sits" in-between event logs and process models. The figure also shows the three types of process mining.

The first type of process mining is *discovery*, i.e., deriving information from some event log without using an a priori model. Based on an event log various types of models may be discovered, e.g., process models, business rules, organizational models, etc.

The second type of process mining is *conformance checking*. Here the event log is used to check if reality conforms to some model. For example, there may be a process model indicating that purchase orders of more than one million Euro require two checks, while in reality this does not happen. Conformance checking may be used to detect deviations, to locate and explain these deviations, and to measure the severity of these deviations.

The third type of process mining, called *extension*, also assumes both a log and a model as input (cf. Figure 2). However, the model is not checked for correctness, instead it is used as a basis, i.e., the model is augmented with some new information or insights. For example, an existing process model could be extended by timing information, correlations, decision rules, etc.

Orthogonal to the three types of mining, there are the three perspectives mentioned before. The *control-flow perspective* focuses on the control-flow, i.e., the ordering of activities. The goal of mining this perspective is to find a good characterization of all possible paths, e.g., expressed in terms of a Petri net or some other notation (e.g., EPCs, BPMN, UML ADs, etc.). The *organizational perspective* focuses on information about resources hidden in the log, i.e., which performers are involved and how are they related. The goal is to either structure the organization by classifying people in terms of roles and organizational units or to show the social network. The *case perspective* focuses on properties of cases. Cases can be characterized by their path in the process or by the originators working on a case. However, cases can also be characterized by the values of the corresponding data elements. For example, if a case represents a replenishment order, it may be interesting to know the supplier or the number of products ordered.

3 Tool Support: ProM

The ProM framework aims to *cover the full process mining spectrum* shown in Figure 2. The current version of ProM provides more than 250 plug-ins. The ProM framework has been developed as a completely plug-able environment and serves as an excellent basis for process mining initiatives.

ProM is the only comprehensive framework supporting a wide range of process mining techniques. Most other tools in this area only focus on a single perspective and/or technique. *Futura Reflect* by Futura Process Intelligence, *BPM|one* by Pallas Athena, *Comprehend* by Open Connect, *Interstage Automated Business Process Discovery and Visualization* by Fujitsu, *Process Discovery Focus* by Iontas, and *Enterprise Visualization Suite* by BusinesScape are some examples of commercial tools that offer some form of process discovery. Of these tools Futura Reflect and BPM|one are more mature as they allow for the discovery of processes with concurrency. Most of the other tools mentioned are only able to discover sequential processes or even require a-priori modeling. Commercial tools typically offer only a small subset of the functionality provided by ProM. However, the emergence of these tools illustrates the practical interest in process mining. For example, Futura Process Intelligence and Pallas Athena have been selected as "Cool Vendor 2009" by Gartner because of their process mining capabilities. Both tools use genetic process mining algorithms developed in the context of ProM [19].

The reader is referred to www.processmining.org to learn more about process mining and to download ProM.

4 TomTom4BPM

In [2], the term *TomTom4BPM* was coined to stress the need for the map-based functionality one can find in navigation systems (e.g., TomTom, Garmin, VDO Dayton, Mio, Magellan, etc.), Google maps, Google Street View, Mashups using geo-tagging (e.g., Panoramio, HousingMaps, FindByClick, etc.). After introducing process mining, we revisit the desired functionalities mentioned in Section 1. Here we are particularly interested in adding innovative functionality to BPMSs.

- As indicated earlier, *good process maps are typically missing* in today's organizations. Clearly, process mining can assist here. Process discovery algorithms [9, 6, 10, 11, 13, 28, 16, 27] are able to extract process maps from event logs. These maps are describing the way things really happened rather than providing some subjective view.
- In Section 1, we indicated that even if process models exist in an explicit form, *their quality typically leaves much to be desired.* Using process mining techniques, one can avoid depicting a "PowerPoint reality" and come closer to the quality of road maps. Moreover, based on historic information, it is possible use intuitive visual metaphors adopted from road maps. For example, we can use intuitive colors and shapes of varying sizes, e.g., the "highways in the process" are emphasized by thick colorful lines and "process dirt roads" are not shown or shown using thin dark lines. The major "cities of a process" can also be emphasized and less relevant activities can be removed. Relevance can be determined based on actual frequencies of activities in logs. Other metrics may be the time spent on activities or the costs associated with them. ProM's *Fuzzy Miner* [16] can discover processes from event logs and offers such visualizations.
- Most process modeling languages have a static decomposition mechanism (e.g., nested subprocesses) without the ability to *seamlessly zoom in or zoom out like in a navigation system or Google maps.* ProM's *Fuzzy Miner* [16] allows for such a seamless zoom. Note that, while zooming out, insignificant activities and paths are either left out or dynamically clustered into aggregate shapes (e.g., streets and suburbs amalgamate into cities).
- When "process maps" are used in an operational sense, they typically *attempt to control the users.* However, when using a car navigation system, the

driver is always in control, i.e., the road map (or TomTom) is not trying to "control" the user. The goal of an information system should be to provide directions and guidance rather than enforcing a particular route. ProM's *Recommendation Engine* [23] learns from historic data and uses this to provide recommendations to the user. This way the workflow system can provide more flexibility while still supporting the user. This is comparable to the directions given by a navigation system.
– A navigation system continuously shows a clear overview of the current situation (i.e., location and speed). Moreover, *traffic information* is given, showing potential problems and delays. Since process mining results in a tight connection between events and maps, it is easy to project dynamic information on process maps. Ideas such as the ones presented Figure 1 have their counterparts in BPMSs, e.g., showing "traffic jams" in business processes.
– At any point in time the navigation system is showing the *estimated arrival time*. Existing BPMSs are not showing this information and do not recalculate the optimal process based on new information. ProM provides several so-called *prediction engines* [7, 14] to estimate the remaining flow time of a case. The next section shows an example of an application of the technique described in [7].

In this paper, we cannot present the various techniques supported by ProM in detail. Instead, we only show that event logs can be used to predict the remaining time until completion for running cases.

5 An Example: Case Prediction

As an illustration of the innovative features that can be provided by combining accurate process maps and historic event information, we briefly show an application of *case prediction* [7]. To illustrate the technique presented in [7] and implemented in ProM, we use an event log of municipality taken from a process that deals with objections (i.e., appeals) against the real-estate property valuation or the real-estate property tax. The municipality is using eiStream workflow (formerly known as Eastman Software and today named Global 360) to handle these objections.

The process considered in this case study is called "Bezwaar WOZ", where WOZ ("Waardering Onroerende Zaken") refers to the particular law describing regulations related to real-estate property valuation by municipalities. We used an event log with data on 1882 objections handled by the municipality. The log contains 11985 events and the average total flow time is 107 days while some cases take more than 200 days. Figure 3 shows the distribution of total flow times. The x-axis shows the 1882 cases and the y-axis shows the duration in days. Note that some cases take a very short time while others take much longer, thus making it difficult to predict the remaining time for cases in the system. To measure the quality of predictions, we split the log into a training set (log L_1) and a test set (log L_2). Log L_1 contains 982 cases and log L_2 contains 900 cases.

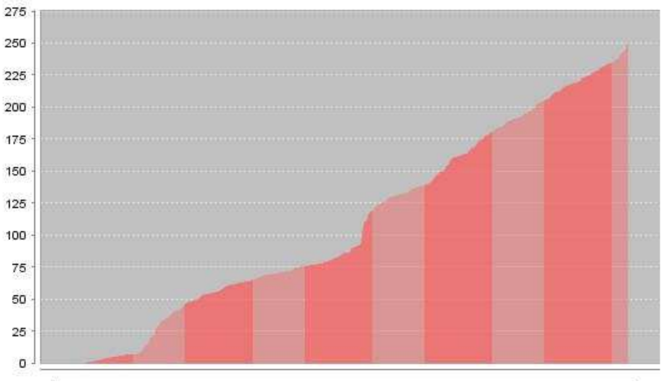

Fig. 3. The distribution of the total flow time of cases extracted using ProM. The x-axis represents the 1882 cases sorted by flow time. The y-axis shows durations in days. Note that some cases almost take 250 days.

The goal is to predict, at any point in time, the remaining processing time of a case. This corresponds to the "estimated arrival time" provided by car navigation systems like TomTom. To do this, we build a so-called *annotated transition system* using the training set (log L_1). Using a variable abstraction mechanism, partial traces are mapped onto states of the transition system. Using historic information, appropriate statistics are collected per state.

Figure 4 shows an annotated transition system obtained using a particular abstraction (see [7] for other abstractions). If one is interested in the remaining time until completion of a particular case c, then the partial trace σ_c of this case is mapped onto a state s_c. Based on s_c a lookup is done in Figure 4 resulting in a prediction t_c^p, e.g., for a case where two particular steps have been executed, the predicted remaining time until completion is $t_c^p = 20.5$ days. Afterwards, it is possible to measure what the actual quality of this estimate. For example, if the real remaining time until completion turns out to be $t_c^r = 25.7$, then the error is $|t_c^p - t_c^r| = 5.2$ days.

If we use the annotated transition system shown in Figure 3 (which was derived from L_1) to predict the remaining time until completion before/after every event in L_2, then the Mean Average Error (MAE) is 17.129 days. Given the fact that there are huge variations in flow times and that the average flow time is 107 days (cf. Figure 3), this is a spectacular performance. For processes which less variation, it is possible to make even better predictions. To put the MAE of 17.129 days into perspective, it is interesting to compare the performance of the annotated transition system shown in Figure 3 with the simple heuristic of always estimating half of average total flow time (i.e., 53.5 days). The MAE of this heuristic is 61.750 days. Hence, the performance of the technique presented in [7] is much better than simple heuristics. It is quite remarkable that one can predict the remaining time until completion so accurately. This shows that

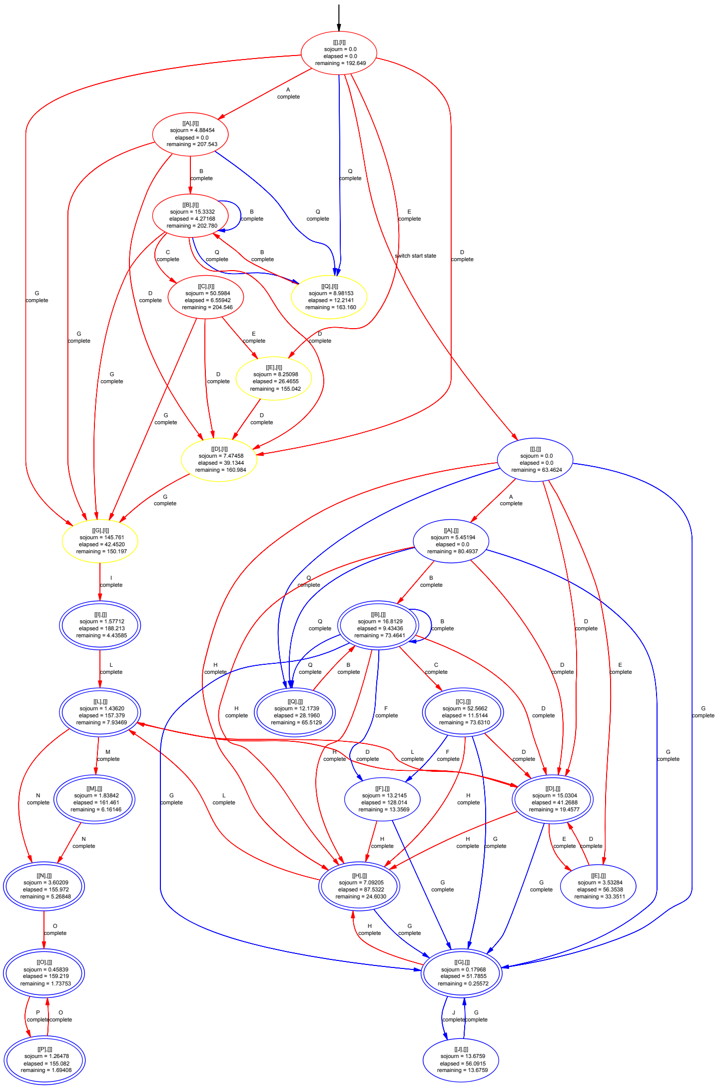

Fig. 4. An annotated transition system extracted from event log L_1. The transition system and its annotations are not intended to be readable and the activity names have been obfuscated. The transition system is learned from an event log containing information about 982 cases (objections against the real-estate property valuation/tax). Per state, historic information is used to make a prediction. For example, for the top state the predicted time until completion is 192 days, for the bottom-left state the predicted time until completion is 1.69 days, and for the bottom-right state the predicted time until completion is 13.67 days. The Mean Average Error (MAE) is 17.129 days when this annotated transition system is evaluated using another log (L_2) containing event data on 900 other objections.

using process mining techniques one can realize TomTom-like functionality like the estimated arrival time.

6 Related Work

Since the mid-nineties several groups have been working on techniques for process mining [9, 6, 10, 11, 13, 28, 16, 27], i.e., discovering process models based on observed events. In [8] an overview is given of the early work in this domain. The idea to apply process mining in the context of workflow management systems was introduced in [10]. In parallel, Datta [13] looked at the discovery of business process models. Cook et al. investigated similar issues in the context of software engineering processes [11]. Herbst [17] was one of the first to tackle more complicated processes, e.g., processes containing duplicate tasks. Most of the classical approaches have problems dealing with concurrency. The α-algorithm [9] was the first technique taking concurrency as a starting point. However, this simple algorithm has problems dealing with complicated routing constructs and noise (like most of the other approaches described in literature). In the context of the ProM framework [3] more robust techniques have been developed. The heuristics miner [27] and the fuzzy miner [16] can deal with incomplete, unbalanced, and/or noisy events logs. The two-phase approach presented in [6] allows for various abstractions to obtain more useful models. It is impossible to give a complete review of process mining techniques here, see www.processmining.org for more pointers to literature.

The approaches mentioned above focus on control-flow discovery. However, when event logs contain time information, the discovered models can be extended with timing information. For example, in [25] it is shown how timed automata can be derived. In [20] it is shown how any Petri net discovered by ProM can be enriched with timing and resource information.

The above approaches all focus on discovering process models based on historic information and do not support users at run-time. The recommendation service of ProM learns based on historic information and uses this to guide the user in selecting the next work-item [23]. This is related to the use of case-based reasoning in workflow systems [26]. In the context of ProM two prediction approaches are supported: [7] and [14]. The prediction service presented in [14, 12] predicts the completion time of cases by using non-parametric regression. The prediction service presented in [7] (used in Section 5) is based on annotated transition systems and uses the abstractions defined in [6]. Also related is the prediction engine of Staffware [24, 22] which is using simulation to complete audit trails with expected information about future events. This particular approach is rather unreliable since it is based on one run through the system using a copy of the actual engine. Hence, no probabilities are taken into account and there is no means of "learning" to make better predictions over time. A more refined approach focusing on the transient behavior (called "short-term simulation") is presented in [21].

The limitations related to the representation and visualization of process models mentioned at the beginning of this paper became evident based on experiences gathered in many process mining projects. It seems that the "map metaphor" can be used to present process models and process information in completely new ways [16, 18]. In the context of YAWL [4, 18], we showed that it is possible to show current work items on top of various maps. Work items can be shown on top of a geographic map, a process model, a time chart, an organizational model, etc. In the context of ProM, we have used the "map metaphor" to enhance the so-called Fuzzy Miner [16]. As presented in [16], four ideas are being combined in ProM's Fuzzy Miner to draw maps of process models.

- *Aggregation:* To limit the number of information items displayed, maps often show coherent clusters of low-level detail information in an aggregated manner. One example are cities in road maps, where particular houses and streets are combined within the city's transitive closure.
- *Abstraction:* Lower-level information which is insignificant in the chosen context is simply omitted from the visualization. Examples are bicycle paths, which are of no interest in a motorists map.
- *Emphasis:* More significant information is highlighted by visual means such as color, contrast, saturation, and size. For example, maps emphasize more important roads by displaying them as thicker, more colorful and contrasting lines (e.g., motorways).
- *Customization:* There is no one single map for the world. Maps are specialized on a defined local context, have a specific level of detail (city maps vs highway maps), and a dedicated purpose (interregional travel vs alpine hiking).

7 Conclusion

The paper suggests *using process mining to create accurate and interactive business process maps* for the management of business processes. The maps can be accurate because they are no longer based on outdated or subjective information, but on facts recorded in event logs. By establishing a close connection between event logs and such maps, it is possible to project information dynamically and let the user interact with such business process maps. Using ProM some of the desired TomTom functionality has been realized and there is a huge innovation potential for today's BPMSs. Using "TomTom4BPM" we can realize truly intelligent information systems.

To make things a bit more concrete, we presented a particular example of such functionality using a new method for predicting the "future of a running instance". Given a running case, our prediction approach allows answering questions like "When will this case be finished?", "How long does it take before activity A is completed?", "How likely is it that activity B will be performed in the next two days?", etc. This corresponds to the functionality we know from modern car navigation systems that give an estimate for the remaining driving time.

Essentially for all of this is that we have high-quality business process maps. Unfortunately, the quality of today's process models leaves much to be desired and the situation is comparable to cartographic information decades ago. Problems with the first navigation systems showed that incorrect maps result in systems that are not very usable. Therefore, the ability to extract maps from event logs using process mining is crucial.

Some people may argue that business processes are less stable than infrastructures consisting of roads, intersections, and bridges. Therefore, it is much more difficult to provide accurate business process maps. This is indeed the case. However, this illustrates that a continuous effort is required to keep business process maps up to date. Process mining can be used for this. Moreover, by recording and analyzing event logs on-the-fly, it is possible to offer more flexibility without loosing sight of the actual processes. Therefore, the need to enforce rigid processes is removed and, like in the context of a car navigation system, the "driver is in control" rather than some archaic information system.

8 Acknowledgements

This research is supported by EIT, NWO-EW, SUPER, and the Technology Foundation STW. We would like to thank the many people involved in the development of ProM (see references).

References

1. W.M.P. van der Aalst. Business Process Management Demystified: A Tutorial on Models, Systems and Standards for Workflow Management. In J. Desel, W. Reisig, and G. Rozenberg, editors, *Lectures on Concurrency and Petri Nets*, volume 3098 of *Lecture Notes in Computer Science*, pages 1–65. Springer-Verlag, Berlin, 2004.
2. W.M.P. van der Aalst. TomTom for Business Process Management (TomTom4BPM). In J. Gordijn, editor, *Proceedings of the 21st International Conference on Advanced Information Systems Engineering (CAiSE'09)*, Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2009.
3. W.M.P. van der Aalst, B.F. van Dongen, C.W. Günther, R.S. Mans, A.K. Alves de Medeiros, A. Rozinat, V. Rubin, M. Song, H.M.W. Verbeek, and A.J.M.M. Weijters. ProM 4.0: Comprehensive Support for Real Process Analysis. In J. Kleijn and A. Yakovlev, editors, *Application and Theory of Petri Nets and Other Models of Concurrency (ICATPN 2007)*, volume 4546 of *Lecture Notes in Computer Science*, pages 484–494. Springer-Verlag, Berlin, 2007.
4. W.M.P. van der Aalst and A.H.M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4):245–275, 2005.
5. W.M.P. van der Aalst, H.A. Reijers, A.J.M.M. Weijters, B.F. van Dongen, A.K. Alves de Medeiros, M. Song, and H.M.W. Verbeek. Business Process Mining: An Industrial Application. *Information Systems*, 32(5):713–732, 2007.
6. W.M.P. van der Aalst, V. Rubin, B.F. van Dongen, E. Kindler, and C.W. Günther. Process Mining: A Two-Step Approach to Balance Between Underfitting and Overfitting. *Software and Systems Modeling*, 2009.

7. W.M.P. van der Aalst, M.H. Schonenberg, and M. Song. Time Prediction Based on Process Mining. BPM Center Report BPM-09-04, BPMcenter.org, 2009.
8. W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. *Data and Knowledge Engineering*, 47(2):237–267, 2003.
9. W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.
10. R. Agrawal, D. Gunopulos, and F. Leymann. Mining Process Models from Workflow Logs. In *Sixth International Conference on Extending Database Technology*, pages 469–483, 1998.
11. J.E. Cook and A.L. Wolf. Discovering Models of Software Processes from Event-Based Data. *ACM Transactions on Software Engineering and Methodology*, 7(3):215–249, 1998.
12. R. Crooy. Predictions in Information Systems: A process mining perspective. Master's thesis, Eindhoven University of Technology, Eindhoven, 2008.
13. A. Datta. Automating the Discovery of As-Is Business Process Models: Probabilistic and Algorithmic Approaches. *Information Systems Research*, 9(3):275–301, 1998.
14. B.F. van Dongen, R.A. Crooy, and W.M.P. van der Aalst. Cycle Time Prediction: When Will This Case Finally Be Finished? In R. Meersman and Z. Tari, editors, *Proceedings of the 16th International Conference on Cooperative Information Systems, CoopIS 2008, OTM 2008, Part I*, volume 5331 of *Lecture Notes in Computer Science*, pages 319–336. Springer-Verlag, Berlin, 2008.
15. M. Dumas, W.M.P. van der Aalst, and A.H.M. ter Hofstede. *Process-Aware Information Systems: Bridging People and Software through Process Technology.* Wiley & Sons, 2005.
16. C.W. Günther and W.M.P. van der Aalst. Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics. In G. Alonso, P. Dadam, and M. Rosemann, editors, *International Conference on Business Process Management (BPM 2007)*, volume 4714 of *Lecture Notes in Computer Science*, pages 328–343. Springer-Verlag, Berlin, 2007.
17. J. Herbst. A Machine Learning Approach to Workflow Management. In *Proceedings 11th European Conference on Machine Learning*, volume 1810 of *Lecture Notes in Computer Science*, pages 183–194. Springer-Verlag, Berlin, 2000.
18. M. de Leoni, W.M.P. van der Aalst, and A.H.M. ter Hofstede. Visual Support for Work Assignment in Process-Aware Information Systems. In M. Dumas, M. Reichert, and M.C. Shan, editors, *International Conference on Business Process Management (BPM 2008)*, volume 5240 of *Lecture Notes in Computer Science*, pages 67–83. Springer-Verlag, Berlin, 2008.
19. A.K. Alves de Medeiros, A.J.M.M. Weijters, and W.M.P. van der Aalst. Genetic Process Mining: An Experimental Evaluation. *Data Mining and Knowledge Discovery*, 14(2):245–304, 2007.
20. A. Rozinat, R.S. Mans, M. Song, and W.M.P. van der Aalst. Discovering Simulation Models. *Information Systems*, 34(3):305–327, 2009.
21. A. Rozinat, M.T. Wynn, W.M.P. van der Aalst, A.H.M. ter Hofstede, and C. Fidge. Workflow Simulation for Operational Decision Support Using Design, Historic and State Information. In M. Dumas, M. Reichert, and M.C. Shan, editors, *International Conference on Business Process Management (BPM 2008)*, volume 5240 of *Lecture Notes in Computer Science*, pages 196–211. Springer-Verlag, Berlin, 2008.

22. B. Schellekens. Cycle Time Prediction in Staffware. Master's thesis, Eindhoven University of Technology, Eindhoven, 2009.
23. H. Schonenberg, B. Weber, B.F. van Dongen, and W.M.P. van der Aalst. Supporting Flexible Processes Through Recommendations Based on History. In M. Dumas, M. Reichert, and M.C. Shan, editors, *International Conference on Business Process Management (BPM 2008)*, volume 5240 of *Lecture Notes in Computer Science*, pages 51–66. Springer-Verlag, Berlin, 2008.
24. Staffware. *Staffware Process Suite Version 2 – White Paper*. Staffware PLC, Maidenhead, UK, 2003.
25. S.E. Verwer, M.M. de Weerdt, and C. Witteveen. Efficiently learning timed models from observations. In L. Wehenkel, P. Geurts, and R. Maree, editors, *Benelearn*, pages 75–76. University of Liege, 2008.
26. B. Weber, W. Wild, and R. Breu. CBRFlow: Enabling Adaptive Workflow Management Through Conversational Case-Based Reasoning. In *Advances in Case-Based Reasoning*, volume 3155 of *Lecture Notes in Computer Science*, pages 434–448. Springer-Verlag, Berlin, 2004.
27. A.J.M.M. Weijters and W.M.P. van der Aalst. Rediscovering Workflow Models from Event-Based Data using Little Thumb. *Integrated Computer-Aided Engineering*, 10(2):151–162, 2003.
28. J.M.E.M. van der Werf, B.F. van Dongen, C.A.J. Hurkens, and A. Serebrenik. Process Discovery using Integer Linear Programming. In K. van Hee and R. Valk, editors, *Proceedings of the 29th International Conference on Applications and Theory of Petri Nets (Petri Nets 2008)*, volume 5062 of *Lecture Notes in Computer Science*, pages 368–387. Springer-Verlag, Berlin, 2008.
29. M. Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer-Verlag, Berlin, 2007.