Facilitating Process Analysis through Visualising Process History: Experiences with a Dutch Municipality

Massimiliano de Leoni[1], Joos C.A.M. Buijs[1], Wil M. P. van der Aalst[1,2], and Arthur H. M. ter Hofstede[2,1]

[1] Eindhoven University of Technology, Eindhoven, The Netherlands
{m.d.leoni,j.c.a.m.buijs,w.m.p.v.d.aalst}@tue.nl
[2] Queensland University of Technology, Brisbane, Australia
a.terhofstede@qut.edu.au

Abstract. Nowadays vast quantities of data are stored as a result of the operation of software systems and devices. The analysis of this data can provide valuable insights. In the field of Business Process Management, event logs may provide valuable information for business process improvement. This is the realm of process mining, an area which has provided many analysis techniques over the past decade. Despite the abundance of process mining techniques, it remains a challenge to provide results that are understandable by domain experts. Discovered process models are often perceived as abstract and static. Conformance checking techniques provide detailed results that are only understandable for process analysts. Therefore, we propose an approach to dynamically visualize event data on intuitive "maps". States of the process are visualized on a collection of maps thus resulting in sequences of "photographs" of the process under investigation. By replaying the event log using such visualizations we can create a collection of "process movies". Our visualisation approach has been implemented in ProM and allows for any type of "map" as long as activity instances can be associated to map coordinates. Moreover, the approach has been evaluated in collaboration with a Dutch municipality.

1 Introduction

As a result of increased automation and increased capacity of storage, more and more data is recorded by today's software systems and devices. The McKinsey Global Institute (MGI) estimates that enterprises globally stored more than 7 exabytes of new data on disk drives in 2010, while consumers stored more than 6 exabytes of new data on devices such as PCs and notebooks [16]. The amount of data recorded in various domains has been growing exponentially, thereby following Moore's law. While the availability of large amounts of data is an enabler for various forms of analysis, the sheer quantity and diversity of this data creates new challenges.

The area of process mining (see e.g. [1]), which emerged a little over a decade ago, is concerned with the analysis of data as stored in event logs. In this relatively short timespan it has proven to be capable of providing deep insight into process-related problems that contemporary enterprises face. Through the application of process mining, organisations can discover the processes as they are conducted in reality, they can check whether certain practices and regulations were really followed and they can gain insight into bottlenecks, resource utilisation, and other performance-related aspects of processes.

Despite the fact that the field of process mining has shown itself to be a valuable addition to the BPM landscape, dealing with vast collections of data still remains a challenge. This is primarily caused by the fact that human beings need to be able to understand the analysis results and this requires domain knowledge and creativity. While visualisation is a powerful tool to aid human comprehension, the field of *visual analytics*, a term coined by Jim Thomas in [21], aims to provide a tight integration between visualisation and automated analysis with the goal of deepening this comprehension. This field comprises a number of disciplines and combines elements from human-computer interaction, geo-spatial and temporal data processing, data analysis, and statistics [12, 13]. A recent definition states that "Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets" [12]. A starting point of this paper is the belief that the application of techniques from the field of visual analytics can play a significant role in overcoming the challenges related to the analyses of large collections of (process) data.

In [15] a *map* metaphor was used to aid people in the selection of activities to perform. A "map" could e.g. be a geographical map, a timeline, or an organisational chart and activity instances are positioned on this map according to their properties. In addition, the colour of a dot representing an activity instance is determined its status or *distance*, e.g., the colour of a dot representing an activity instance may indicate how familiar a resource is with performing the activity (familiarity), how close the activity is to its deadline (urgency), or to many resources it is offered (popularity). The approach focussed on showing the current state of the information system at run-time. This approach can easily be extended for a-posteriori analysis: using the information stored in event logs, it is possible to replay the past history and build the states the system went through. Hence, for each "map", a sequence of different "photographs" can be build, showing how activities were projected on the map in each of these states. If, for each map, the constructed sequence of photographs is played in succession, one obtains a different "movie". These movies, one per map, provide analysts and domain experts with helicopter view of the past execution history seen from different angles. In [3] a framework was proposed that exploits the map metaphor to show a summary of an event log in the form of a movie.

The contributions of this paper are twofold. First, an implementation for the framework provided in [3] is described. This implementation has been realised as a plug-in for the ProM open-source process mining framework (www.processmining.org). Second, a detailed evaluation of the framework with stakeholders from a Dutch municipality has been conducted. This evaluation is concerned with determining whether the map metaphor is understood and can help provide insight to these stakeholders.

The paper is organised as follows. Section 2 positions our work with respect to the literature, highlighting the limitations that exist in relevant state-of-the-art work. Section 3 summarises the formalization of the framework proposed in [3] in order to make this paper self-contained. Then, Section 4 discusses the implementation of this framework developed for the ProM environment. Section 5 describes a case study where the approach has been applied in the context of an information system of a Dutch municipality, whereas Section 6 reports the methodology used during the evaluation with end users, i.e. process designers and analysts working at the municipality, along with the results of this evaluation. Finally, Section 7 concludes the paper by summarizing the results obtained and identifying future directions of work.

2 Related Work

The approach presented in this paper builds on two emerging disciplines: process mining [1, 23] and visual analytics [12, 21]. As a general comment one can observe that while many techniques have been developed for process mining, and also for data mining or statistical analysis, they often do not, or insufficiently, take visualisation aspects into account. Conversely, one can observe that the research community in the area of information visualisation (see e.g. [7] or [20]) has not focussed on process-related aspects.

As far as visualization-related work in the field of business processes is concerned, there exist a number of papers (e.g., [4, 10, 18, 19]) which employ different metaphors and also mash-up approaches to represent the state of a process at run-time. The approach described in [15] elaborates on these ideas and captures a process state as a map with a colouring scheme used for the activities representing their status or their distance. This approach though is aimed at providing run-time support for activity selection and not at providing support for the analysis of the history of a process.

The term "visual analytics" was coined in [21]. This reference reviews the early work in this field. A comprehensive up-to-date reference is provided in [12]. Examples of recent significant research in the area of visual analytics can be found in document analysis [17], financial analysis [14, 24] and geo-spatial object analysis [5]. In [11] pioneering work is reported where visual analytics is applied to the field of data mining.

While the scope of visual analytics does not exclude the visualization of process-related data most of the work in this area is either quite generic or tailored to a particular application. A systematic approach to making visual analytics "process aware" is missing [2]. Temporal aspects (e.g. relating to concurrency or causality) need to be visualised in a clear and consistent manner. To our knowledge, the approach of the so-called *Fuzzy Animations* [9] is the only approach that can be considered to be process-aware. It focuses on providing a graphical user interface where the past states extracted from the event log are projected onto a process model which is automatically derived from the events in the log. It is a valuable approach though quite specific: it only focuses on control-flow aspects and visualization is limited to process graphs.

3 The Framework

To make this paper self-contained, we first provide a brief summary of the framework presented in [3] (with some small adaptations). This framework is concerned with the mapping of an event log to a "movie" summarizing process history with the aim of facilitating process analysis.

The events in an event log can be sorted chronologically and subsequently be replayed in that order. The occurrence of an event makes the system enter a particular state. Hence, by replaying all events in the event log in chronological order, it is possible to rebuild a process history, i.e. the sequence of states the system went through. Each state can be represented as a configuration of activities on a set of maps of choice. In order to define how states can be represented on a map, we have to choose an image to use as map, define how activities are to be positioned on this map and, more generally, agree on a colour scheme to represent their state. Such an annotated map can be seen as a "photograph" and, thus, a process history can be visualised as a sequence of such photographs, that together form a "movie".

An *activity instance* is the execution of a certain activity in a certain case and formally it can thus be represented as a pair (a_{name}, a_{cid}), with $a_{name} \in A$ the name of the activity (A is the universe of activities) and $a_{cid} \in C$ the case (C is the universe of case identifiers). Processes also access and modify data. Let V be the universe of variable names, a *process data variable* is a pair (v_{name}, v_{cid}) where $v_{name} \in V$ and $v_{cid} \in C$. These variables can take on different values in different cases and also within the same case as time progresses.

Our framework only provides visualizations for activity instances that have been created but are not yet concluded. Such activity instances can be in a number of states: they can be *scheduled* when they have been created but not yet been assigned to a resource, they can be *assigned* when they have been assigned to a resource but have not yet started execution, *executing* when work on them has commenced, and *suspended* when work on them has been temporarily halted. We will use the set Z to capture the various states an activity instance can be in, $Z = \{Scheduled, Assigned, Executing, Suspended, Concluded\}$. An activity instance is referred to as *active* if it is a state which is not concluded. To simplify the formalisation we explicitly include the state *Concluded* capturing the state where work on an activity has been finalised.

Definition 1 (Event). *Let U be the universe of values that attributes can take on. An event e is a tuple (a_{name}, cid, t, z, P) where:*

- *$a_{name} \in A$ is an activity name;*
- *cid is a case identifier;*
- *t is the timestamp when event e occurred;*
- *$z \in Z$ is the state to which the corresponding activity instance moves,*
- *$P : V \not\rightarrow U$ is an assignment of values to attributes. Function P is partial since not every event has an associated value for all process variables.*

We use the following functions to access the constituent elements of an event $e = (a_{name}, cid, t, z, P)$, $activity(e) = a_{name}$, $case(e) = cid$, $timestamp(e) = t$, $state(e) = z$ and $properties(e) = P$. The latter function will be overloaded and $properties(e, v) = P(v)$. Moreover, given a function f, $dom(f)$ represents the domain of f.

Definition 2 (System State). *A system state $S = (\alpha, \nu)$ consists of:*

- *a function $\alpha : (A \times C) \not\rightarrow Z$ where $\alpha(a_{name}, a_{cid}) = z$ denotes that activity instance (a_{name}, a_{cid}) is in state $z \in Z$ when the system state is S.*

– *a function* $\nu : (V \times C) \not\rightarrow U$ *where* $\nu(v_{name}, v_{cid}) = v_{value}$ *denotes that variable* (v_{name}, v_{cid}) *has value* v_{value} *in system state* S.

Here, we assume that each pair (a_{name}, a_{cid}) is unique. As result, it is not possible to have, within the same case, multiple instances of the same activity a_{name} that are concurrently active. Instances of the same activity in a loop are treated as a single activity instance that, multiple times, enters and leaves the same states. However, focussing on showing the states entered by the system and not the state transitions, loops do not actually cause problems, since the resulting system state would be the same in either situations.

Similar to existing algorithms for conformance checking [1], this framework is based on the principle of replay. Events in the log are replayed, as to determine a posteriori the state changes that they have triggered. Nevertheless, our notion of replay is slightly different as we are concerned with state changes. This is formalised in the next definition. First, let us define the operator $\oplus$ for function overriding. Let f be an n-ary function, a function $f' = f \oplus (\phi_1, \ldots, \phi_n, \phi_{n+1})$ is such that $f'(\phi_1, \ldots, \phi_n) = \phi_{n+1}$ and, for all $\Phi \in dom(f) \setminus \{(\phi_1, \ldots, \phi_n)\}$, $f'(\Phi) = f(\Phi)$. The definition of $\oplus$ can be extended for tuple sets, by iterative applying the definition to all tuples in the set.

Definition 3 (Replaying of events). *Let* $S = (\alpha, \nu)$ *be the current state during replay and* e *be the next event to replay. Replaying* e *causes the current state* S *to change to state* $S' = (\alpha', \nu')$*. This change is denoted as* $S \xrightarrow{e} S'$*, where*

$$\alpha' = \alpha \oplus \{(activity(e), case(e), state(e))\}$$
$$\nu' = \nu \oplus \{(v, case(e), properties(e, v)) \mid v \in dom(properties(e))\}$$

The initial state from which replaying starts is $S_0 = (\alpha_0, \nu_0)$ where $dom(\alpha_0) = \varnothing$ and $dom(\nu_0) = \varnothing$. Replaying is used to reconstruct the *execution history*.

Definition 4 (Execution History). *Let* $\langle e_1, \ldots, e_n \rangle$ *be the sequence of events in an execution log ordered by timestamp, i.e. for every* $i < n$ *and for every* $i < j \leq n$, $timestamp(e_j) \geq timestamp(e_i)$*. Let* $S_0 \xrightarrow{e_1} S_1 \xrightarrow{e_2} \ldots \xrightarrow{e_n} S_n$ *be the sequences of states visited when replaying the event log. An* execution history *is a sequence of pairs* $H =< (S_1, t_1), \ldots, (S_n, t_n) >$ *where* (S_i, t_i) *denotes that the system entered state* S_i *at time* $t_i = timestamp(e_i)$.

From here on, we will not further provide a formalisation, for this we refer the reader to [3], but only provide a brief textual summary.

Analysts can choose from a variety of maps. For each of these maps it should be defined how activity instances should be positioned. These positions may depend on the values of process variables. It is possible that the evaluation of a position of an activity instance yields coordinates that are outside the map of choice. Those activity instances are, then, shown in a separate list. An example of this could be a Gantt-chart where activity instances which are overdue cannot be shown on the timeline. There can also be activity instances of which the position cannot be computed. These are represented in another list.

Dots on a map may overlap. These are joined to form *aggregated dots*, since otherwise some of them would be invisible. The center of the aggregated dot is computed

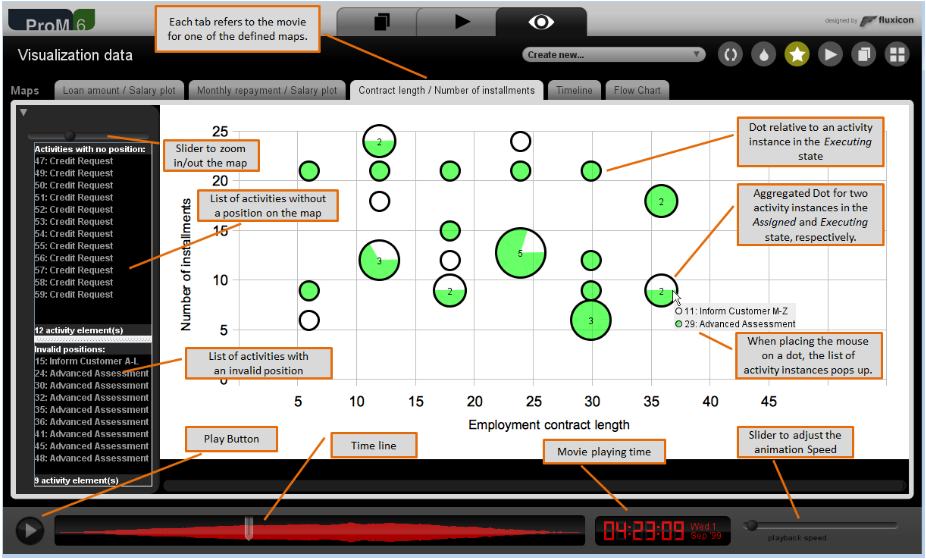


Fig. 1. A tool screenshot showing the visualisation of the output, i.e. the set of different "movies" ready to be replayed. The button on the bottom left can be clicked to start/stop the movie.

as the average of the centres of all dots that have been merged. The diameter of an aggregated dot grows logarithmically with the number of constituent activity instances. Aggregated dots are divided in as many slices as there are constituting dots and each slice is filled with the colour of the dot that it corresponds to. Slices filled with the same colour are also grouped together.

The logarithmic increase in size is a graphical metaphor which allows users to quickly and easily perceive the number of amalgamated dots. Nevertheless, in a few maps, this amalgamation can cause interpretation problems. Indeed, there can be a cascading effect if the diameter grows with the number of activity instances involved, as the growth of an aggregated dot can lead to the dot overlapping with other dots, which, then, need to be incorporated, as well. This may lead to dots involving many activity instances which are positioned in a place possibly quite far away from the original position of some of these activity instances. As this may lead to interpretation issues, the framework allows end users to turn off the amalgamation feature for specific types of maps.

4 Implementation of the Framework

The approach described in this paper has been implemented as a software plug-in of ProM, an open-source framework for the implementation of process mining tools in a standardised environment [22]. To avoid "reinventing the wheel" each time a new process mining algorithm is invented, ProM is developed as a "plugable" environment where new process mining algorithms can be implemented as plug-ins. The look-and-feel of these plug-ins is then the same though their functionality may differ. Plug-ins require a number of input objects and produce one or more output objects. These input objects could, for example, be event logs or output objects of other plug-ins. In this way, one can define a tool chain.

The *LogOnMapReplay* plug-in requires two input objects: an event log and a collection of specifications of maps of interest. Event logs need to be converted to the XES format[3] in order to be used in ProM 6. The XES format is the standard adopted by the IEEE Task Force on Process Mining [23]. A map specification is an XML file that encodes a set M of available maps, where each map consists of a name and a URL of the corresponding image, and a collection of position functions, one for each map.

Figure 1 shows a sample visualisation of the output of the *LogOnMapReplay* plug-in. The bottom part of the screenshot shows the widgets for controlling the playing of the "movie". From left to right, this area contains *(i)* the button to start/stop the movie, *(ii)* a box which shows the relative progression of the movie and an indication of the number of active activity instances along the way, *(iii)* a clock showing the playing time, as obtained through the timing information present in the log, and *(iv)* a slider to adjust the speed of the movie. The box shows a wave in which the x-axis represents time and the y-axis the number of active activity instance. The box also contains a slider that users can move along the x-axis to directly go to a certain snapshot. The pane at the centre of the screenshot is organized through several tabs, each associated with a different map. By selecting a tab, the corresponding map is brought to the front and it is shown the state of the workflow at that specific point in time, in terms of the activity instances that are active, their state and their positions. While playing the animation, users can change the "movie" they are watching in case they wish to consider a different type of map.

The top list on the left-hand side of the screenshot enumerates the activity instances which are not to be positioned on the map, while the bottom list enumerates those activity instances whose position is invalid. The log used as a basis for the screenshot of Figure 1 was artificially generated. This log records events where activity instances can be in the state *Scheduled*, *Executing* or *Concluded* (hence not *Assigned* or *Suspended*). The log concerns a process for handling loan requests and deciding whether these requests are granted or rejected.

The map shown in the screenshot of Figure 1 is a two-dimensional chart where the x-axis refers to the length of the employment contract of an applicant (in months) and the y-axis to the number of instalments in which the loan applied for should be paid off. Activity instances are positioned according to the values of these two properties of the case, the length of the employment contract and the number of instalments. Activity instances referring to cases where the applicant has a permanent position are put on the "no position" list, whereas those activity instances part of cases where contract length information is not provided are put in the "invalid" list. Placing the mouse on a dot, whether it is aggregated or not, reveals the, one or more, activity instances that belong to this dot (see the dot on the bottom right in the map). Aggregated dots may have white and green parts, whose relative sizes show the ratio of activity instances part of that dot which are assigned versus those which are executing. The number of activity instances part of an aggregated dot is shown in the centre of the dot. Note that activity instances

[3] See http://www.xes-standard.org/

above the diagonal $y = x$ in Figure 1 need to be monitored carefully as these refer to cases where repayment is less likely to occur (i.e., the number of instalments is more than the length of the contract).

5 A Real Scenario for a Dutch Municipality

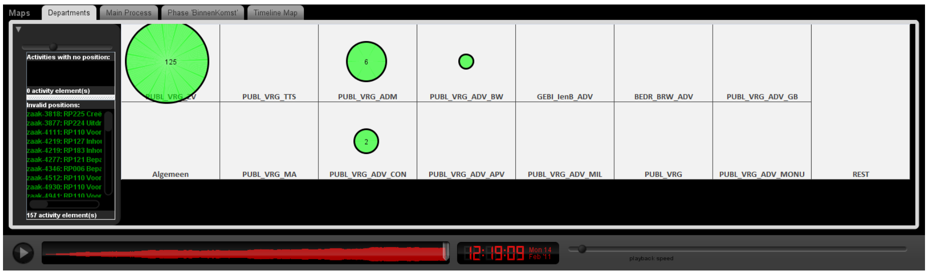
In October 2010, the Netherlands introduced a new law (the so-called "WABO" law) for processing building related permits. This new law subsumes about 25 types of permits and notices that were handled separately before. Typically when an application is received by the municipality, it is checked for completeness and admissibility by the departments involved (building department, environmental department, fire department, etc.). Then, a confirmation of receipt and possibly a request for additional data is sent to the applicant. Once the period given to the applicant to provide the additional data has expired, a decision is made based on the information available. This results in an official document and an invoice to the applicant. During this whole process the allowed term for handling the application can be extended once by the municipality. Furthermore, if the applicant failed to provide the requested missing information within the time period, the application can be rejected.

We have applied the framework on projecting logs on maps in the context of the CoSeLoG project, which analyses process variants within municipalities and the development of services that support these different process variants.[4] At the end of 2010, as a result of the introduction of the "WABO" law, one of the municipalities participating in the CoSeLoG project decided to adapt the process to the changed circumstances and adopt a new case management system. After more than one year of enacting the new process, the municipality was interested to investigate the effects of the procedural changes. In particular, the municipality was interested in discovering possible issues encountered during the execution, such as bottlenecks and inefficient usage of time and resources. In order to discover these issues, the *LogOnMapReplay* plug-in was used in interactions with various stakeholders from the municipality.

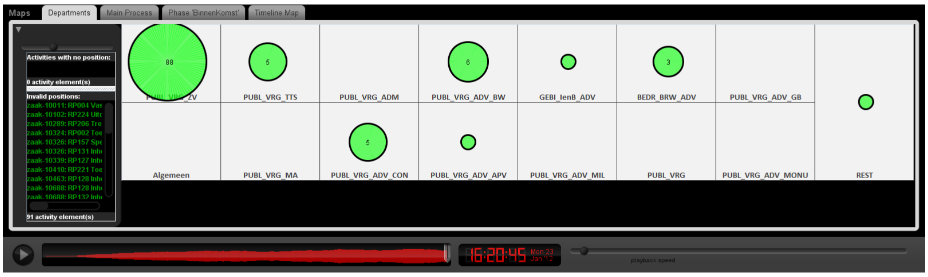
To this end, we extracted an event log from the municipality's information system and analysed it using different log visualizations available in ProM (e.g. the Dotted Chart). After consultation with the process designers and analysts of the municipality on a number of occasions, we defined a number of relevant maps. Interested readers can visit the web page at http://www.processmining.org/online/logonmaps to download some videos that illustrate the application of these maps to this case study. In this paper, we show some "photographs" of these "movies", namely some of those that we consider to be the most important. However, these may be misleading as the value of our approach can only be experienced by playing the movies and watching how the state projection onto maps dynamically changes.

The first relevant map concerns an organizational map that highlights the different groups participating in the process. In total there are twenty four groups involved in handling the permit process. Through a log analysis we found that 14 groups are actively involved in the process and 10 are only occasionally involved. These 14 groups are

[4] See http://www.win.tue.nl/coselog/



(a) State Projection on February 14, 2011



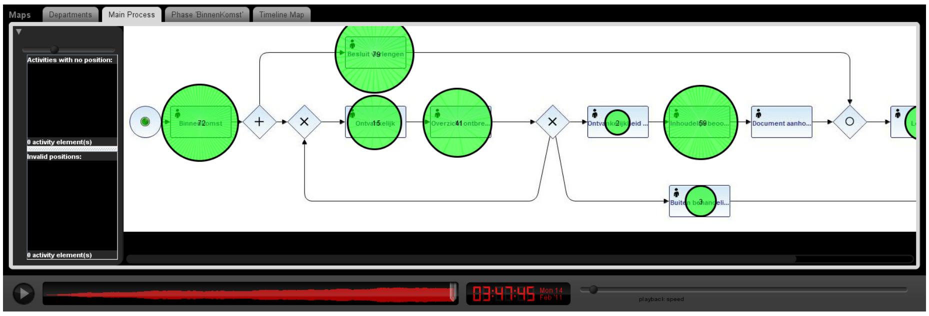
(b) State Projection on January 23, 2012

Fig. 2. State Projections on a Department Map at two Different Moments.

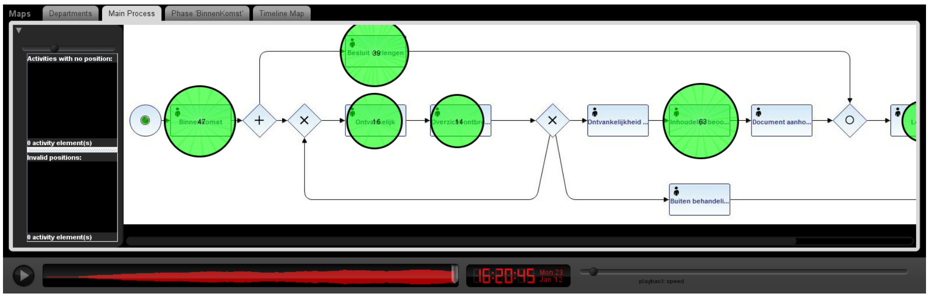
shown as separate areas on the map while the other 10 groups are aggregated into a single area labeled with *Rest*. Figure 2 shows a possible way of projecting the state of a process onto such a map. Each active activity instance is positioned onto the area representing the department group to which it is assigned. For some active activity instances, no information about the department group is available, so their position is considered to be invalid and, hence, they are listed in the bottom list on the left-hand side.

Figure 2 shows state projections related to two different points in time: (a) February 14, 2011 and (b) January 23, 2012. In both of these “photographs”, it is evident that most activity instances are assigned to *PUBL_VRG_ZV* (an acronym for the “Case Handler” group): 125 and 88 activity instances respectively. The other groups are assigned a relatively small number of activity instances, which suggests these groups play a supporting or advising role in the process while the case handler group performs the majority of actions. Nevertheless, in late January 2012, compared to middle February 2011, there are fewer activity instances assigned to the group “Case Handler” and more to the other groups, hence the work is more spread across the groups.

The municipality’s case management system closely follows a process model for handling requests for building permits. This process model is composed of more than one hundred activities; hence, it cannot be directly used as a map, as it would be difficult to interpret. Therefore, we decided to group activities into phases and obtained an aggregated process model which is more understandable when used as a map (see



(a) State projection on February 14, 2011



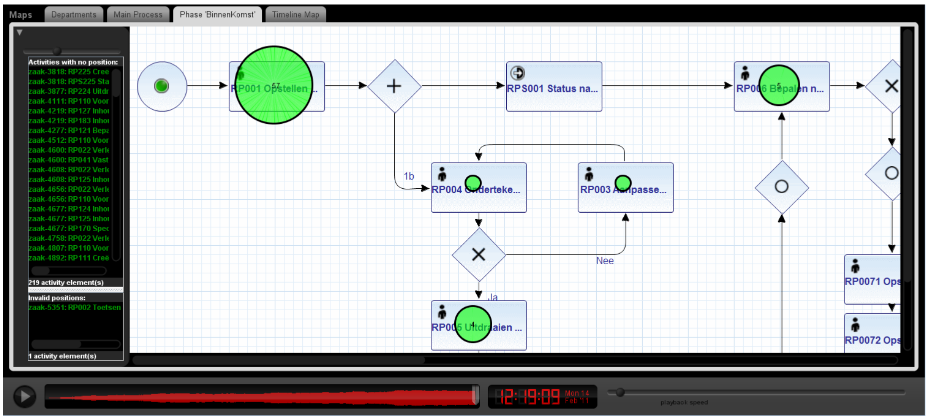
(b) State projection on January 23, 2012

Fig. 3. State projections on a process phase map at two different moments.

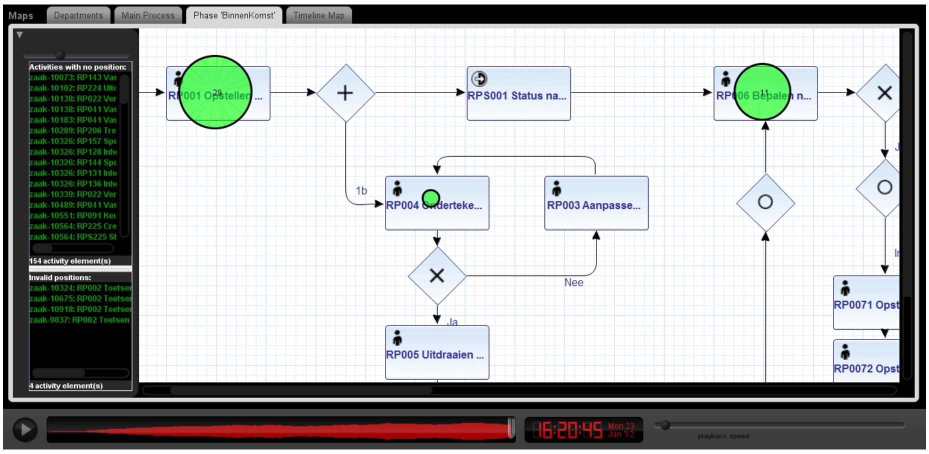
Figure 3). At the same time, in order to drill down into the different phases, it is possible to generate different "sub-process maps" for each phase. As a proof of concept, we have taken the first phase, i.e., *Binnenkomst* ("Arrival" in English), from the original process model used within the municipality. This phase is concerned with handling new incoming permit requests. Figure 4 shows part of the corresponding process map.

In Figure 3, activity instances are projected onto the phase they are a part of. For this map, the lists on the left-hand side are always empty: all activity instances are projected and their position is never invalid. Comparing the situation on February 14, 2011, with the situation on January 23, 2012, we see similar instance distributions over the various phases, e.g., it is evident that some phases, e.g. *Binnenkomst*, have more dots than other phases. However, we can also see that the amount of work at the beginning of 2011 exceeds to work-in-progress at the beginning of 2012. By replaying the "movie" for this map, it can be seen that activities are quickly handled in certain phases: dots are just shown for short periods.

The decomposition of the first phase *Binnenkomst* into individual activities allows one to gain more detailed insights. Now activity instances are not associated with any position if they are not part of the first phase and those instances are enumerated in the top left list. The instances for activity *RP002 Toetsen ontvangstbevestiging* ("Check Confirmation of Receipt") are enumerated in the bottom left list as their positions are



(a) State projection on February 14, 2011

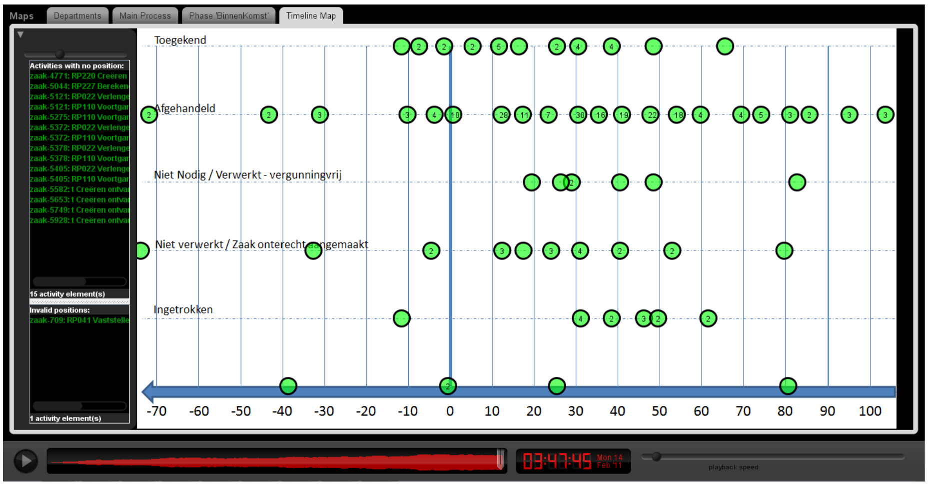


(b) State projection on January 23, 2012

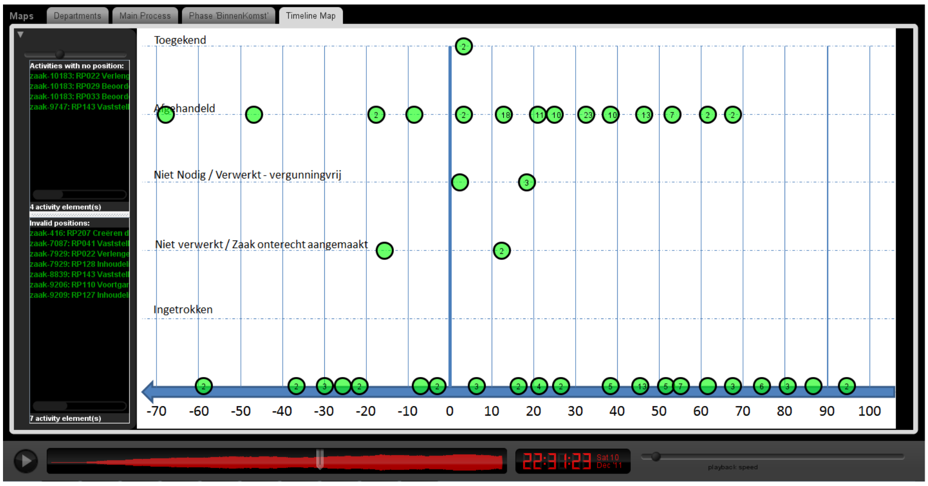
Fig. 4. State projections on a process map for the phase *Binnenkomst* ("Arrival") at two different moments. In order to preserve readability, we only show part of the map in the screenshots.

considered invalid. The reason for this is that activity *RP002 Toetsen ontvangstbevestiging* is not part of the municipality's process model, but instances of this activity have been recorded in the event log. The reverse is also true, some (mandatory) activities from the process model do not appear in the event log. This is a first indication that the actual execution of the process is not fully compliant with the process model and a further analysis is necessary to investigate the degree of compliance between the real process executions and the process model (e.g., using conformance-checking techniques).

A comparison of the two "photographs" shown in Figure 4 for February 14, 2011, and January 23, 2012, illustrates the recurrent patterns of behaviour that can be seen when watching the full movie. Work tends to pile up for the initial activity *RP001*



(a) State projection on February 14, 2011



(b) State projection on December 10, 2011

Fig. 5. State projections on a timeline map at two different moments.

Opstellen Ontvangstbevestiging ("Create Confirmation of Receipt"). This suggests that requests are typically processed until reaching the end of the phase before new requests are looked at, rather than they are worked in parallel shortly after their arrival, which would allow giving a quick feedback to applicants. This observation is also supported by the fact that (relatively) few dots are present on the other activities of this phase. Moreover, when playing the "movie" for this map, it is easy to notice that some activities do not have active instances in the 3 months from November 2011 to January 2012 inclusive (e.g., *RP003 Aanpassen ontvangstbevestiging* which can be translated to "Adapt Confirmation of Receipt") while there were corresponding active instances in the beginning of 2011. This suggests that the actual process model has evolved over time and that some process fragments are no longer used. This example shows that our approach can be used to show *concept drift* [6], the process is changing while being analysed and is not in steady state.

Another map is shown in Figure 5. In this map, activity instances are positioned according to the number of remaining days before the case deadline (x-axis) and the outcome of the case (y-axis). Activity instances for cases for which the final outcome is not known are positioned on the x-axis (e.g., cases that are not yet completed). Thus, in this animation dots move from right to left as the case approaches the deadline. It should be noted that the diameter of aggregated dots does not increase to avoid the cascading effect discussed at the end of Section 3 (overlapping aggregated dots clutter together into event larger dots). Activity instances that are extremely overdue (i.e. more than seventy days over the deadline) are enumerated in the list of those with invalid positions. Activity instances of which the associated cases have deadlines more than one hundred days into the future do not have an associated position.

Figure 5(a) shows the projection of the process state on February 2011: most activity instances are for cases that are going to complete with the result *Afgehandeld* (in English "Successfully Processed"). Only a few activity instances refer to cases that are overdue, which shows that permit requests are usually finalised within time. Looking at Figure 5(b), the projection of the process state at the end of 2011 shows a somewhat different picture with more activity instances belonging to cases which are extremely overdue (7 versus 1). In addition, one can see that there are fewer cases, of which the status is other than *Afgehandeld*. There are also more activity instances on the x-axis (hence with unknown status), but this is due to the fact that the event log contains the events until February 2012 and, hence, at that time, the outcome of many cases being performed in December 2011 was still unknown. Comparing the "photographs" from February 2011 and from December 2011, it is quite clear that there are significantly more cases in the pipeline on February 2011.

6 Evaluation of the Approach with End Users

We performed an evaluation with end-users of the municipality in order to verify whether the approach is valid, useful, and sufficiently intuitive. We presented the "movies" described in Section 5 to employees of the municipality that provided us with the event data. The end-user evaluation was conducted using the well-established *Thinking Aloud* methodology [8] where participants are asked to use the system to execute some tasks and express aloud their impressions. The advantage is that end users can really express what they feel; the drawback is that the environment may be slightly artificial and, hence, affect the results. Two evaluation sessions involved three subjects: a process management specialist, a communication and marketing specialist and a business advisor for customer contacts. We showed the subjects the four different maps described in the previous section and we asked them to formulate conclusions based on what they had seen.

We carefully chose subjects with a different level of knowledge of the specific process and a different level of training in interpreting process models and related information. Due to the distinctive roles of the subjects in the municipality, we were interested in differences in appreciation. For example, the process management specialist was expected to have a broader view on the process and mostly focussed on response times and bottlenecks, whereas the other two subjects were expected to be mostly focussed on the relations with customers.

The first evaluation session was only with the process management specialist. The second evaluation session was with the other subjects, who were also supported by the process management specialist. Both sessions were conducted using the *LogOnMapReplay* plug-in, i.e., the *LogOnMapReplay* plug-in was used to show movies concerning the period from October 2011 to January 2012 (based on the event data provided by the municipality). In both of sessions, we only gave a brief introduction of the framework and of the movies. According to the thinking aloud methodology, no further comments were given, thus letting the subjects draw their own conclusions. Without interfering, we could evaluate the level of understandability of the map metaphor and usefulness of the approach when extracting knowledge from event logs.

Outcomes of a First Evaluation Session with a BPM Specialist. The process management specialist who participated to the evaluation is involved in several projects of process redesign and improvement. Even though he had no detailed knowledge of the specific process, he could easily interpret the meaning of the movies and draw interesting conclusions. Using the aggregated process model map, he immediately noticed how work tends to accumulate in the phases *Binnenkomst* ("Arrival") and *Inhoudelijke beoordeling* ("Assessment of the content"). He also noticed that work is carried out very quickly in other phases, such as *Buiten behandeling stellen* ("Stop handling case"). If we compare these conclusions with ours, which were described in Section 5, they perfectly coincide. The process analyst highly appreciated the presence of an aggregated process model map, which allowed him to have a helicopter view of the process executions. Regarding the detailed process model for phase *Binnenkomst* ("Arrival"), he immediately noticed how, in this phase, work tends to accumulate for the first activity which is about the notification of the reception of the permit requests to applicants. This behaviour does not coincide with what the process management specialist would like to see: the first process activity should be executed soon after a request arrives. By immediately notifying applicants of the request's reception, the service level improves, along with the customers' satisfaction.

Playing the "movie of the department map", the process management specialist immediately noticed that most of the work is assigned to the group *PUBL_VRG_ZV* (acronym for the "case handler" group). Each member of this group is responsible for, and the main actor of, a number of cases. The other groups are only occasionally involved when special analysis is needed to evaluate the permit requests, e.g. checking fire or construction safety. Therefore, it makes perfect sense that this group is assigned a large portion of the activities. He also noticed that some of the other groups are not assigned too many activities. He was also delighted to be able to track the amount of work assigned to every group and evaluate the length of the different work queues.

Indeed, for an optimal utilization of resources, each group should have assigned a reasonable amount of work (i.e. neither too much nor too little). He noticed, however, that the work queues are related to a single process: to have a clearer picture, the event log should have contained cases of multiple processes, i.e. all those which the groups are involved in.

The "movie of the time-line map" allows the process management specialist to draw other conclusions, too. While playing the movie, it was evident to him that many dots disappear when they are very close to the vertical line of deadline. From this, he clearly concluded that municipality employees work driven by deadlines: they give priority to work on resident permits when they are getting close to the deadline. He discussed with us that, ideally, employees should work in a different way and start working on resident permits as soon as they are submitted.

Outcomes of a Second Evaluation Session with Active Process Participants. A week later, we conducted the second evaluation session with two additional subjects that are actively involved in the execution of the process, i.e. a communication and marketing specialist and a business advisor for customer contacts. The process management specialist was also present for support. First, we showed the movie for the maps for the different phases. Immediately the two subjects stated that this map would be very useful for the management of WABO cases. The map clearly indicates the distribution of work among the phases. They could easily recognise the phases where most of work tends to accumulate. Regarding the detailed process model for the phase *Binnenkomst* ("Arrival"), the two subjects noticed that many activities in the phase are rarely executed and most of the work is related to the first activity. In the same way as the process specialist, they observed how cases flowed through the other activities rather quickly. They also found it very useful that, when placing the mouse on a dot, aggregated or not, a list pops up showing detailed information about the activity instance. This provides confidence and allowed the subjects to look up details in their system when needed.

The "movie for the organizational map" also provided new insights. The business advisor for customer contacts quickly noticed that the number of active activity instances handled by groups *PUBL_VRG_ZV* (acronym for the "case handler" group), *PUBL_VRG_ADM* (the "administration" group) and *BEDR_BRW_ADV* ("fire safety advice") fluctuated heavily, whereas the volume of work-in-progress for other groups was much more stable. He could deduce that these groups handled their activities rather quickly. The fact that the subjects could extract this behavioural pattern from such a relatively simple map is quite interesting and shows that it is not strictly necessary to design very elaborated maps to extract knowledge and recurrent behavioural patterns.

Finally, the time-line animation was shown. The fact that many activities were not executed before the deadline, and even some were going over the deadlines, suggested them that there may be a structural problem within the process. Another thing the subjects noted was that, although it is very useful to see a snapshot of the situation at a certain moment in time, it is not possible to easily derive when dots disappear from the maps. The subjects suggested that each dot could leave a trace or a mark for a certain time after it is no longer projected. They were also concerned with the fact that it is not very clear how long a certain dot is projected on a map. Since it is not possible to easily distinguish dots, it is not clear how long they are projected. Indeed, dots can be projected on different positions at different moments in time.

Lesson Learned and Consequent Directions of Improvement. In summary, the subjects involved in the evaluation clearly appreciated the approach and its application to the case study. They could easily draw useful conclusions from the movies provided after only a brief introduction of the framework and the maps. Therefore, the map metaphor was clearly understood and could help the stakeholders to gain a deeper insight into the process performance.

Nonetheless, some problems made the movies' interpretation unnecessarily complex. As mentioned earlier, activity instances currently leave no trace when they disappear. This is particularly confusing when dots are moving and can disappear at multiple positions on the map. In order to address this limitation, we plan to incorporate a new feature: when activity instances are completed, dots gradually fade out until they completely vanish after a certain number of photographs.

Moreover, the stakeholders remarked that it is sometimes unclear how long activity instances are active. Currently, dots are coloured according to the state of the relative activity instance. However, dots could also be coloured according to different properties of the activity instances. We plan to provide different approaches, allowing users to select the property used to colour the dots. For example, in order to make evident how long single activity instances are projected onto the maps, we could use the colour of a dot to show how long the activity instance is active, e.g., white dots are related to activity instances that have recently become active, whereas red dots refer to "old" activity instances, with intermediate shades of red indicating intermediate values. Another problem that subjects have raised is that dots do not clearly show the case of the relative activity instance. The only way to know the case is to place the mouse on the dot. We plan to incorporate an approach where dots are coloured based on the identity of the corresponding case. Dots related to activity instances of the same case are filled with the same colour.

7 Conclusion

Process models can be viewed as geographic maps. However, unlike real maps the quality of process models often leaves much to be desired. Man-made process models tend to be subjective and disconnected from real process executions. Process mining techniques can be used to improve the quality of process maps, e.g., process discovery techniques can be used to automatically derive process models from event data and conformance checking techniques can be used to pinpoint and quantify deviations between model and reality. However, this is not sufficient as process maps are static and do not show the flow of work. Therefore, we developed an approach to visualize process histories in a generic manner. Different "maps" can be used as long as activity instances can be given coordinates on such a map, e.g., an activity instance may be mapped onto a Gantt chart, an organisation chart, a process model, etc. By showing a sequence of "photographs" of the process (i.e., a "movie"), one can see concept drift, compliance problems, moving bottlenecks, etc.

This paper describes an implementation of these ideas using the ProM framework. The implementation is generic and any collection of maps can be used as long as it is possible to map instances onto map coordinates. Moreover, dots on the map can be coloured using properties of the corresponding activity instance. The approach has been evaluated using a case study in the context of the CoSeLoG project. Employees of a Dutch municipality were provided with a set of "movies" based on an event log of the so-called WABO process. They were able to understand these "movies" with little effort and gained new insights by viewing the process from different angles. The case study also showed that there is room for further improvements. For example, the employees of the Dutch municipality expressed the desire to easily derive when activity instances disappear from a map and how long they are active. Now it is difficult to track individual instances that move over the map and potentially disappear at different locations. Another important direction of improvement is concerned with reducing the end-user effort when defining maps and the projection of activity instances on them. Many maps have a broad range of applicability (e.g., a time-line or a process model map) and can be used in many different settings with few or no changes. Therefore, we are currently working on a ProM plug-in that automates the definition of certain types of maps, by self-generating the map picture and the projection of activity instances.

The results in this paper show the value of combining process mining and visual analytics. Process mining results are often perceived to be rather abstract and static. Visual analytics approaches tend to be data-centric rather than process-centric. The combination of both fields may yield innovative process-centric visualizations such as the "process movies" proposed in this paper.

Acknowledgement. The authors would like to thank Atilla Erdodi, MSc student at the University of Amsterdam and the Eötvös Lorànd University in Budapest, who has contributed to the development of the *LogOnMapReplay* plug-in as part of his thesis work partially conducted at Eindhoven University of Technology.

References

1. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer-Verlag (2011)
2. van der Aalst, W.M.P.: Using process mining to generate accurate and interactive business process maps. In: Abramowicz, A., Flejter, D. (eds.) Business Information Systems (BIS 2009) Workshops. Lecture Notes in Business Information Processing, vol. 37, pp. 1–14. Springer-Verlag (2009)
3. van der Aalst, W.M.P., de Leoni, M., ter Hofstede, A.H.M.: Computational Intelligence, chap. Process Mining and Visual Analytics: Breathing Life into Business Process Models. Nova Publisher (2012)
4. Alonso, G., Hagen, C.: Geo-Opera: Workflow Concepts for Spatial Processes. In: SSD'97: Proceedings of the 5th International Symposium on Advances in Spatial Databases. Lecture Notes in Computer Science, vol. 1262, pp. 238–258. Springer (1997)
5. Bak, P., Mansmann, F., Janetzko, H., Keim, D.: Spatio-temporal Analysis of Sensor Logs using Growth Ring Maps. IEEE Transactions on Visualization and Computer Graphics 15, 913–920 (2009)
6. Bose, R.J.C., van der Aalst, W., Zliobaite, I., Pechenizkiy, M.: Handling Concept Drift in Process Mining. In: International Conference on Advanced Information Systems Engineering. Lecture Notes in Computer Science, vol. 6741, pp. 391–405. Springer-Verlag (2011)

7. Chen, C.: Information Visualization: Beyond the Horizon. Springer-Verlag, New York, Inc. (2006)
8. Dix, A., Finlay, J.E., Abowd, G.D., Beale, R.: Human-Computer Interaction. Prentice Hall, 3rd edn. (2003)
9. Günther, C.W., van der Aalst, W.M.P.: Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) International Conference on Business Process Management (BPM 2007). Lecture Notes in Computer Science, vol. 4714, pp. 328–343. Springer-Verlag (2007)
10. Kaster, D., Bauzer-Medeiros, C., da Rocha, H.V.: Supporting Modeling and Problem Solving from Precedent Experiences: The Role of Workflows and Case-based Reasoning. Environmental Modelling and Software 20(6), 689–704 (2005)
11. Keim, D.: Visual Exploration of Large Data Sets. Communications of the ACM 44, 38–44 (2001)
12. Keim, D., Kohlhammer, J., Ellis, G., Mansmann, F. (eds.): Mastering the Information Age: Solving Problems with Visual Analytics. VisMaster, http://www.vismaster.eu/book/ (2010)
13. Keim, D., Mansmann, F., Schneidewind, J., Thomas, J., Ziegler, H.: Visual analytics: Scope and challenges. In: Simoff, S.J., Böhlen, M.H., Mazeika, A. (eds.) Visual Data Mining, pp. 76–90. Springer-Verlag (2008)
14. Keim, D., Nietzschmann, T., Schelwies, N., Schneidewind, J., Schreck, T., Ziegler, H.: A Spectral Visualization System for Analyzing Financial Time Series Data. In: Proceedings of the Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis 2006). pp. 195–200. Eurographics Association (2006)
15. de Leoni, M., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Visual Support for Work Assignment in Process-Aware Information Systems. In: International Conference on Business Process Management (BPM 2008). Lecture Notes in Computer Science, vol. 5240, pp. 67–83. Springer-Verlag (2008)
16. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H.: Big data: The next frontier for innovation, competition, and productivity. Tech. rep., McKinsey Global Institute (MGI) (May 2011)
17. Oelke, D., Ming, C., Rohrdantz, C., Keim, D., Dayal, U., Lars-Erik, H., Janetzko, H.: Visual Opinion Analysis of Customer Feedback Data. In: Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (IEEE VAST 2009). pp. 187–194. IEEE (2009)
18. Schönhage, B., van Ballegooij, A., Elliëns, A.: 3D Gadgets for Business Process Visualization—A Case Study. In: VRML '00: Proceedings of the fifth symposium on Virtual reality modeling language (Web3D-VRML). pp. 131–138. ACM (2000)
19. Schönhage, B., Eliëns, A.: Management Through Vision: A Case Study Towards Requirements of BizViz. In: AVI 2000: Internation Conference of Information Visualisation. pp. 387–392. IEEE Computer Society (2000)
20. Spence, R.: Information Visualization: Design for Interaction. Pearson Education Limited, Harlow, England, 2nd edn. (2006)
21. Thomas, J., Cook, K. (eds.): Illuminating the Path: The Research and Development Agenda for Visual Analytics. IEEE CS Press (2005)
22. Verbeek, H.M.W., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: XES, XESame, and ProM 6. In: Soffer, P., Proper, E. (eds.) Information Systems Evolution. Lecture Notes in Business Information Processing, vol. 72, pp. 60–75. Springer-Verlag (2010)
23. W.M.P. van der Aalst et al.: Process mining manifesto. In: Proceedings of Business Process Management Workshops 2011. LNBIP, vol. 99. Springer Verlag (2012)
24. Ziegler, H., Nietzschmann, T., Keim, D.: Relevance Driven Visualization of Financial Performance Measures. In: Proceedings of the Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis 2007). pp. 19–26. Eurographics Association (2007)