

# Turning Event Logs into Process Movies: Animating What Has Really Happened

Massimiliano de Leoni, Suriadi Suriadi, Arthur H. M. ter Hofstede, Wil M. P. van der Aalst

**Abstract**—Today’s information systems log vast amount of data which contains information about the actual execution of business processes. The analysis of this data can provide a solid starting point for business process improvement. This is the realm of process mining, an area which has provided a repertoire of many analysis techniques. Despite the impressive capabilities of existing process mining algorithms, dealing with the abundance of data recorded by contemporary systems and devices remains a challenge. Of particular importance is the capability to guide the meaningful interpretation of this “ocean” of data by process analysts. To this end, insights from the field of visual analytics can be leveraged. An approach is proposed where process states are reconstructed from event logs and visualised in succession, leading to an animated history of a process. This approach is customisable in how a process state, partially defined through a collection of activity instances, is visualised: one can select a map and specify a projection of activity instances on this map based on their properties. In this paper an implementation of the proposal is described for the open-source process-mining framework ProM along with reporting an evaluation with one of Australia’s largest insurance companies: Suncorp.

**Index Terms**—Business Process Mining, Visual Analytics, Event-log Animation, Process Visualisation



## 1 INTRODUCTION

As a result of increased automation and storage capacity, more and more data is recorded by today’s software systems and devices. The McKinsey Global Institute (MGI) estimated 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 [1]. 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 [2].

In the field of Business Process Management (BPM), so-called process-aware information systems record information about the execution of business processes in event logs. Analysing such event logs has been the driver of the area of *process mining* (see e.g. [3]), which emerged a little over a decade ago. In this relatively short timespan, this discipline 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, check whether certain practices and regulations were really followed and 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 large collections of data still remains a challenge. In fact, although automatic tech-

niques are certainly needed, process analysts need to be guided with regards to where to focus their attention in this “ocean of data”, which automatic techniques to choose for further analysis and how to fine-tune these techniques. To achieve this, one can leverage from the field of *visual analytics*, a term coined by Jim Thomas in [4], which “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” [5].

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 [6] 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 by its status or *distance*, e.g., how close the activity is to its deadline, how long it is being executed. The approach focussed on showing the current state of the information system at run-time. This approach can easily be extended to a-posteriori analysis: using the information stored in event logs, it is possible to replay the history and build the states the system went through. Hence, for each “map”, a sequence of different “photographs” can be built, 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 “process movie”. These movies or animations (one per map) provide analysts and domain experts with a helicopter view of the past execution history seen from different perspectives.

- 
- M. de Leoni and W. M. P. van der Aalst are with Eindhoven University of Technology. M. de Leoni is also with University of Padua.
  - S. Suriadi and A. H. M. ter Hofstede are with Queensland University of Technology

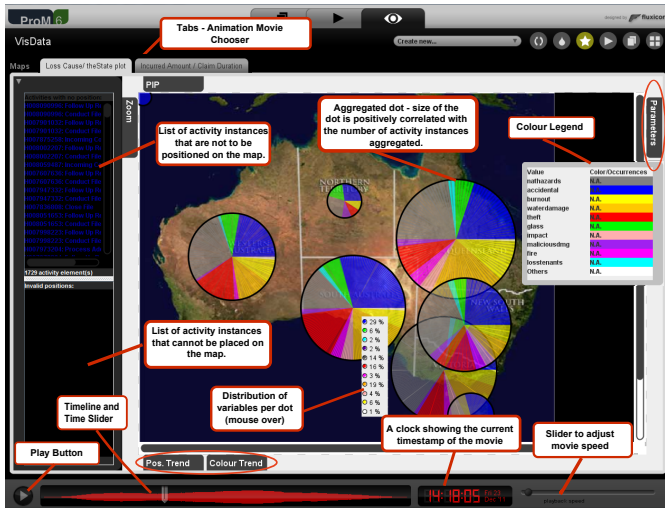


Fig. 1. A screenshot of a process movie referring to the log data of a process enacted in an Australia insurance company to deal with claims. Activity instances are projected onto the Australian state of the claimants.

An example of such a movie is shown in Figure 1 where the event log refers to the execution of process instances to handle insurance claims of an insurance company in Australia. The process map is of a geographic type: each activity instance is projected onto the map as a dot in the time frame when it was being executed as recorded in the log. The dot is positioned on the Australia state of the claimant. At any given point in time, the size of the dots on the map represents the number of activity instances which are projected on to that position. The instances can have different characteristics (e.g., in the figure, relative to the claim's type). Therefore, the areas of dots are also sliced according to the percentage of instances with given characteristics, and different colours are assigned to the slices. Completed activity instances are no longer shown on the map. Thus, Figure 1 is the "photograph" of the activity instances at a particular point in time as shown in the timestamp box (i.e. 23 Dec 2011 14:18:05). The time slider at the bottom of the figure allows users to go back or forth to a certain point in time to view the corresponding photograph. Alternatively, the photographs can be played in a continuous sequence by activating the 'play' button.

In [7] an initial framework was proposed that exploits the map metaphor to show a summary of an event log in the form of an animation. The corresponding implementation was realised as a plug-in for the ProM open-source process mining framework ([www.processmining.org](http://www.processmining.org)). A set of experiments was conducted with some stakeholders from a Dutch municipality to obtain some quick initial feedback [8]. As a result of the feedback, the framework was adjusted to cope with the problems that were identified. In fact, significant extensions and modifications of the initial framework and of the accordant implementation were needed. This paper reports on the resulting framework, its corresponding ProM plug-in,

and a new validation effort, this time with subjects from Suncorp, one of Australia's largest insurance companies. This evaluation is concerned with determining whether the map metaphor is understood and can help to provide meaningful insights.

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 discusses the adjusted framework. Then, Section 4 provides some details of the implementation of this framework realised as a plug-in of the ProM environment. Section 5 describes a case study illustrating the approach in the context of insurance claims processing for homes in Suncorp. Section 6 starts with reporting the feedback from stakeholders of a Dutch municipality followed by the changes that were required to cope with the problems pointed out. Then, the section describes the evaluation of the adapted framework done in collaboration with Suncorp. Finally, Section 7 concludes the paper by summarising 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* [3] and *visual analytics* [5], [4]. As a general comment, one can observe that while many techniques have been developed for process mining, data mining and 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. [9] or [10]) has not focussed on process-related aspects.

There exists a number of research works (e.g., [11], [12], [13]) on aspects related to visualisation in the field of business process management. These works employ different metaphors and also mash-up approaches to represent the state of a process at run-time. The approach described in [6] 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 characteristics. 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 [4]. This reference reviews the early work in this field. A comprehensive more recent reference is provided in [5]. Examples of recent significant research in the area of visual analytics can be found in document analysis [14], financial analysis [15], [16] and geo-spatial object analysis [17]. In [18] pioneering work is reported where visual analytics is applied to the field of data mining. Another interesting work is [19] which concerns displaying multiple time series without aggregation. The above body of work uses static representations for capturing time dependencies, i.e. images that summarise the analysis of time-oriented data. When larger data sets come into play, static representations show their limits as they require large screens

to represent the time axis. Conversely, dynamic representations (i.e., using the physical time) show their power when the focal point is to analyse the data over time. A detailed up-to-date survey is reported in [20], where it is clear that there is no significant research work that uses dynamic representations: “A frequent goal is to integrate data from multiple time stems in a single image”. Often, time abstraction is used for this purpose. Unfortunately, to visualise process-related data, temporal aspects are of crucial importance as they are related to concurrency and causality of the activities being performed. Therefore, abstracting time would relegate the time dimension to second-class status.

Most of the work in the visual analytics area is tailored to a particular application or to a particular visualisation. This is also confirmed by [21]: “To our knowledge, there exists no visualisation framework that [...] provides a broader selection of possible representations. We think that an open framework fed with pluggable visual and analytical components for analyzing time-oriented data is useful. Such a framework will be able to support multiple analysis tasks and data characteristics, which is a goal of Visual Analytics”. In fact, the configuration of different maps allow one to plug different representations, thus supporting multiple views.

To our knowledge, only the approach of the so-called *Fuzzy Animations* [22] 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 visualisation is limited to process graphs.

### 3 THE FRAMEWORK

In this section, we define a formal framework that captures the mapping of event logs to “movies”. Through the resulting movies the history of a process can be visualised in different contexts, thus facilitating its analysis. The framework is derived from the framework presented in [7], where the main difference is that a richer notion of state is adopted. The notion of state as presented here allows one to exploit temporal information when defining the positioning of activity instances.

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 activity instances on a map of choice. In order to define how states can be represented on a map, we have to choose an image for that map and to define the positioning of activity instances as dots on that image. 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”. To convey more information, each dot can be filled with a colour, where this colour depends on the value of a characteristic of choice of the corresponding activity instance.

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

Our framework only provides visualisations 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 which an activity instance can be in,  $Z = \{Scheduled, Assigned, Executing, Suspended, Concluded\}$ . An activity instance is referred to as *active* if it is in a state that is an element of  $Z$  except for the state *Concluded*.

*Definition 1 (Event)*: Let  $U$  be the universe of values that variables can take. An event  $e$  is a tuple  $(at, cid, t, z, P)$  where:

- $at \in A$  is an activity type;
- $cid$  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;
- $P : V \dashrightarrow U$  is an assignment of values to variables.

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  $e = (at, cid, t, z, P)$ ,  $activity(e) = at$ ,  $case(e) = cid$ ,  $timestamp(e) = t$ ,  $state(e) = z$  and  $properties(e) = P$ . The latter function will be overloaded and  $properties(e, vn) = P(vn)$ . Moreover, given a function  $f$ ,  $dom(f)$  represents the domain of  $f$ .

*Definition 2 (System State)*: Let  $T$  be the universe of possible timestamps. A system state  $S = (\alpha, \nu, \tau^s)$  consists of:

- a function  $\alpha : (A \times C) \dashrightarrow Z$  where  $\alpha(at, cid) = z$  denotes that activity instance  $(at, cid)$  is in state  $z \in Z$ ;
- a function  $\nu : (V \times C) \dashrightarrow U$  where  $\nu(vn, cid) = v_{value}$  denotes that variable  $(vn, cid)$  has value  $v_{value}$ ;
- a function  $\tau^s : (A \times C) \dashrightarrow T$  where  $\tau^s(at, cid) = t$  denotes that activity instance  $(at, cid)$  started at timestamp  $t$ .

Due to our definition of activity instance, as a pair consisting of activity name and case identifier, it is not possible to distinguish between different instances of the same activity within the same case. Such instances may arise from the occurrence of loops in a process model.

### 3.1 Creation of the Sequence of States

Similar to existing algorithms for conformance checking [3], this framework is based on the principle of replay. Events in the log are replayed to determine, a posteriori, the sequence of states that the system has gone through. In order to formalise this notion for our framework, let us first define the overriding operator  $\oplus$ . Let  $f$  be a function, a function  $f' = f \oplus (\vec{x}', y)$  is defined by  $f'(\vec{x}') = y$  and  $f'(x) = f(x)$  for all  $x \in \text{dom}(f) \setminus \{\vec{x}'\}$ . The definition of  $\oplus$  can be extended to tuple sets, by iteratively applying the definition to all tuples in the set (noting that the order in which the elements are chosen is not important).

*Definition 3 (Replaying of events):* Let  $S_i = (\alpha_i, \nu_i, \tau_i^s)$  be the current state during replay and  $e$  be the next event to replay. Replaying  $e$  causes the current state  $S_i$  to change to state  $S_{i+1} = (\alpha_{i+1}, \nu_{i+1}, \tau_{i+1}^s)$ . This change is denoted as  $S_i \xrightarrow{e} S_{i+1}$ , where

$$\begin{aligned} \alpha_{i+1} &= \alpha_i \oplus ((\text{activity}(e), \text{case}(e)), \text{state}(e)) \\ \nu_{i+1} &= \nu_i \oplus \{((v, \text{case}(e)), \text{properties}(e, v)) \mid v \in \text{dom}(\text{properties}(e))\} \\ \text{and if } \text{state}(e) &\neq \text{Executing then } \tau_{i+1}^s = \tau_i^s \\ \text{otherwise } \tau_{i+1}^s &= \tau_i^s \oplus ((\text{activity}(e), \text{case}(e)), \text{timestamp}(e)) \end{aligned}$$

The initial state from which replaying starts is  $S_0 = (\alpha_0, \nu_0, \tau_0^s)$  where  $\text{dom}(\alpha_0) = \text{dom}(\nu_0) = \text{dom}(\tau_0^s) = \emptyset$ . Replaying is used to reconstruct the *execution history*.

*Definition 4 (Execution History):* Let  $\langle e_1, \dots, e_n \rangle$  be the sequence of events in an execution log ordered by timestamp, i.e. for every  $1 \leq i < j \leq n$ ,  $\text{timestamp}(e_j) \geq \text{timestamp}(e_i)$ . Let  $S_0 \xrightarrow{e_1} S_1 \xrightarrow{e_2} \dots \xrightarrow{e_n} S_n$  be the sequence of states visited when replaying the event log. An *execution history* is a sequence of pairs  $H = \langle (S_1, t_1), \dots, (S_n, t_n) \rangle$  where  $(S_i, t_i)$  denotes that the system entered state  $S_i$  at time  $t_i = \text{timestamp}(e_i)$ .

### 3.2 Mapping States onto Maps

Activity instances are visualised as dots on a map. By not fixing the type of map, but allowing this choice to be configurable, different types of relationships can be shown thus providing a deeper insight into the context of the work that was performed. Many types of maps can be thought of: geographical maps (e.g., the map of a university's campus), process schemas, organisational diagrams, Gantt charts, etc. Naturally, one can also make highly specialised maps to suit a particular purpose. The positioning of an activity instance may vary across different maps. When the use of a certain map is envisaged, the location of activity instances at runtime on this map should be captured through a formal expression specified at design time.

*Definition 5 (Position function):* Let  $M$  be the set of maps of interest. For each available map  $m \in M$ , there

exists a partial function that returns a pair of expressions for each activity type.

$$\text{position}_m : A \not\rightarrow \text{Expr}(V \cup \{\mathcal{T}, t\}) \times \text{Expr}(V \cup \{\mathcal{T}, t\})$$

where  $\text{Expr}(X)$  is the domain of all expressions that use some of the variables in  $X$ . For each activity instance  $ai = (at, cid)$  and each map  $m$ ,  $\text{position}_m$  returns a pair of expressions. The evaluation of these expressions, over a state  $S$ , returns a pair of coordinates  $(x, y)$  which is the position of  $ai$  on map  $m$  at state  $S$ .

More specifically, variables  $\mathcal{T}$  and  $t$  are used to incorporate references in time: they are used to represent the starting time of activity instances and the current time of replay, respectively. Let  $\xi : X \not\rightarrow U$  be a value assignment of a subset of the variable names in  $X$ . We define  $\text{eval}$  as a function which, given an expression  $f \in \text{Expr}(X)$  and a value assignment  $\xi$ , yields an integer number:

$$\text{eval}[f](\xi) = c$$

where  $c \in \mathbb{Z}$ .

Given a map  $m \in M$ , a state  $S_i = (\alpha_i, \nu_i, \tau_i^s)$ , an activity instance  $ai = (at, cid) \in \text{dom}(\alpha_i)$ , then  $\text{position}_m(at) = (f'_{m,a}, f''_{m,a})$ . The coordinates of  $ai$  on map  $m$  for state  $S_i$  at a given timestamp  $\bar{t}$  is:

$$\text{coord}_m(ai)|_{S_i} = (\text{eval}[f'_{m,at}](\xi_{ai}), \text{eval}[f''_{m,at}](\xi_{ai}))$$

where  $\xi_{ai}(vn) = \nu_i(vn, cid)$ ,  $\xi_{ai}(\mathcal{T}) = \tau_i^s(ai)$  and  $\xi_{ai}(t) = \bar{t}$ .

For example, consider a loan request process where each instance corresponds to a different request. The applicant's monthly income and the requested loan amount are stored in variables *income* and *loan*. One can define a cartesian map  $c$  where every activity instance is associated with a distinct dot whose  $x$  and  $y$  coordinates are determined by the values of these variables. Assume that the maximum values of *income* and *loan*, as seen in the log, are 150000 and 10000 respectively. Also assume that the maximum  $x$  and  $y$  coordinate values on to which a dot can still be properly displayed on map  $c$  are 800 and 600 respectively. To ensure that an activity instance with maximum *income* and *loan* values can be properly displayed, we can define a position function such that  $\text{position}_c(at) = (\text{income} \cdot \frac{800}{150000}, \text{loan} \cdot \frac{600}{10000})$ .

The projection of a state  $S_i = (\alpha_i, \nu_i, \tau_i^s)$  onto a map  $m$  is the projection of activity instances  $ai \in \text{dom}(\alpha_i)$  onto  $m$  at position  $\text{coord}_m(ai)|_{S_i}$ . As can be seen from the definition, the function  $\text{position}_m$  is partial and hence not all activity instances in the state are mapped. This may be because it simply is not meaningful. However, there may also be activity instances in  $\text{dom}(\alpha_i)$  that are not mapped as some of the variables in the position function do not have a value. Another reason for an activity instance not to be projected onto a given map is that its coordinates are invalid, i.e. falling outside that map (for example because the  $x$  or the  $y$  coordinate are negative). Given a map, activity instances that have invalid coordinates or none at all need to be visualised differently from instances with valid coordinates. To this

end, each map is associated with two lists of activities instances, one enumerating the activity instances that do not have coordinates and one enumerating those which have invalid ones.

As mentioned before, the dots representing activity instances can be filled with colour. This allows for richer visualisation as one can take the value of a variable of choice into account. Currently, different colouring schemes are proposed, which are based (i) on the state of the activity instances, (ii) on the characteristics of the case of the activity instance or (iii) on the age of the activity instances.

*Based on the state of the activity instance.* When this scheme is used, an activity instance  $ai$  is coloured according to  $\alpha_i(ai)$  where  $S_i = (\alpha_i, \nu_i, \tau_i^s)$  is the current state. In particular, for activity instances that are in the scheduled, allocated, executing or suspended state, we fill the relative dot with white, cyan, green or black, respectively. Of course, activity instances that are concluded or not scheduled are not represented on a map.

*Based on the characteristics of the case of the activity instance.* When this scheme is used, the end user chooses one of the variables  $vn \in V$  present in the event log. The value of the selected variable determines which colour is used to fill the dots of the activity instances in the case. Let  $S_n = (\alpha_n, \nu_n, \tau_n^s)$  be the last state in the execution history. The dot corresponding to an activity instance  $ai = (at, cid) \in dom(\alpha_i)$  is coloured according to  $\nu_n(vn, cid)$ , i.e. the last value assigned to variable  $vn$  for the case  $cid$ . In particular, the 15 most commonly occurring variable values are associated with the 15 non-white colours of the 16-colour EGA palette<sup>1</sup>. The white colour is excluded as it is used to represent all other variable values. The colour of dots for a particular case should not change during an animation. Otherwise, one can easily lose track of dots. Therefore, the visualisation approach chosen is to ignore any changes to a variable and to just use its final value. This then means that one should not choose a variable for visualisation purposes whose value can change during the execution of a case. In addition, the choice of variable should also be informed by the ability to use it as a meaningful classifier of cases.

*Based on the age of the activity instance.* In this approach, the dots are coloured according to the age of the activity instance, i.e. the amount of time that has elapsed since the instance was started. The colour white is associated with activity instances that just started. As time progresses, the colour of instances that have not completed becomes closer and closer to red. Let  $S_i = (\alpha_i, \nu_i, \tau_i^s)$  be the state at time  $t$ , then the age of an activity instance  $ai = (at, cid) \in dom(\alpha_i)$  is computed as follows:

$$age(ai) = \exp^{-\frac{\ln(2)(t - \tau_i^s(ai))}{MET(at)}}$$

where  $MET(at)$  is the average of the time that was taken to complete instances of activity type  $at$ . For each activity instance  $ai$ ,  $age(ai)$  is always between 0 and 1. If  $t = \tau_i^s(ai)$ , i.e. activity instance  $ai$  was just started,

$age(ai) = 1$ . Value  $age(ai)$  decreases exponentially as  $ai$  ages. When  $t - \tau_i^s(ai) = MET(at)$ ,  $age(ai) = 0.5$ . An established approach is used to map  $age(ai)$  to a colour: the ‘‘Fire Colour Pallet’’ [10]. The colour ranges in intensity from a bright white ( $age(ai) = 1$ ) through yellow, orange ( $age(ai) \sim 0.5$ ), brown, and then to black (as  $age(ai) \approx 0$ ).

In order to deal with dots that may overlap on a certain map, they are represented transparently. The colour of areas of overlap is determined by the colours of the individual dots involved. Unfortunately, this may lead to confusion in some cases as it may be hard to correctly interpret the resulting colour, but the advantage of this approach is that dots whose area is completely covered by one or more other dots still remain visible. If the centres of dots coincide, then the dots involved are merged to form bigger dots in order to avoid that dots whose sizes and centre positions are identical can no longer be visually distinguished from each other. The diameter of such dots grows according to the number  $n$  of activity instances involved. Li et al. [23] conducted an analysis with a number of subjects where they found that quantities represented as circles are most intuitively perceived when the circle grows as a power of 0.4. Applying this observation to our case, i.e. when joining  $n$  dots of different activity instances, the diameter of the resulting dot is computed as  $n^{0.4}$ . The amalgamated dots are also divided in as many slices as there are constituting dots, and each slice is filled with the colour of the dot which it corresponds to.

Another important feature is concerned with handling activities instances when they are going to disappear. If a dot suddenly disappears between two consecutive photographs, end users would not notice that the corresponding activity instance is completed, especially when many dots are visualised at the same time on the maps. Therefore, we have introduced a fading effect: if a dot is going to disappear in  $x$  photographs, it starts fading out (i.e., becomes transparent). Value  $x$  can be customised by an end user on the fly while playing the movie. As the number of photographs in which a dot is going to disappear becomes smaller, the fading effect becomes more pronounced, till the dot completely vanishes.

## 4 IMPLEMENTATION OF THE FRAMEWORK

Figure 2 shows the architecture of the implementation. The yellow component is implemented as a stand-alone Java application whereas the red components are implemented as plug-ins of ProM, an open-source ‘‘pluggable’’ framework for the implementation of process mining tools in a standardised environment (<http://www.promtools.org>). 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 chain of plug-ins’ invocations.

The core software is the *Log-On-Map Replayer* plug-in which takes a map-specification file and an event log

1. [http://en.wikipedia.org/wiki/Enhanced\\_Graphics\\_Adapter](http://en.wikipedia.org/wiki/Enhanced_Graphics_Adapter)



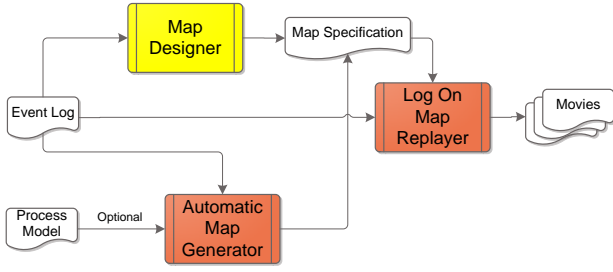


Fig. 2. The architecture of the implementation.

as input. Each map-specification file consists of a set of available maps (i.e. the map name and the URL where the map image can be retrieved) with corresponding position functions (Definition 5), one definition for each map. The plug-in employs the framework defined in Section 3 and generates for each available map a sequence of photographs. Playing such a sequence of photographs in succession yields a movie. As previously mentioned, each photograph captures the state of the process as it existed at a certain point in time. The graphical user interface provides controls to select one of the available movies and put it in focus, play/stop that movie, or go to a specific moment in time in that movie.

Map specifications can be drawn through a Java standalone application, the *Map Designer*. It allows process analysts to load images (e.g., PNG or JPG) to use as maps and to define how to project activity instances onto those maps. To do so, analysts can simply drag and drop an activity type onto the map and place it at the position of interest. This way, the activity type's position is *statically* defined and applies to all instances of that type. Alternatively, analysts can define a position as *dynamic*. The position of an activity instance is then defined in terms of the state of the process instance involved. As far as the implementation is concerned, the state is encoded as an XML document and the position function is defined as an XQuery over this document. Section 5 illustrates the interface of the *Map Designer* application and the *Log On Map Replayer* plug-in through the case study.

There exists a number of potential maps that can be applied to a wide range of scenarios. For example, a process model can serve as a map where activity instances are projected onto the icons representing the corresponding activity types. Hence, as long as the positions of the activity icons are known, this type of map can be used in different scenarios and the corresponding positioning function can be automatically generated. This concept of facilitating the generation of maps with little effort required by users has led to the implementation of a second plug-in of ProM, the *Automatic-Map-Generator*. It takes an event log as input and optionally a process model and produces a map, intended as the background image, and the corresponding position function. Currently, three types of maps can be

generated automatically: Cartesian, Deadline/Timeline, Process-Model. When generating a map of these types, users need to define the size of the image through specifying the width  $w$  and the height  $h$ .

*Cartesian Map.* This type of map takes inspiration from the Cartesian coordinate system. The projection of activity instances is determined by choosing two numerical variables,  $v_x$  and  $v_y$ , to derive the values of  $x$  and  $y$  coordinates from. Let  $x_{\max}$  ( $x_{\min}$ ) and  $y_{\max}$  ( $y_{\min}$ ) be the maximum (minimum) values for  $v_x$  and  $v_y$  as present in the event log. The position function for a Cartesian map  $c$  for instances of an activity type  $at \in A$  is defined as  $position_c(at) = (x'_{at}, y'_{at})$  where

$$x'_{at} = \frac{v_x - x_{\min}}{x_{\max} - x_{\min}} \cdot w \quad \text{and} \quad y'_{at} = \frac{v_y - y_{\min}}{y_{\max} - y_{\min}} \cdot h.$$

For example, suppose that the end user chooses variables *payout* and *amount* as  $v_x$  and  $v_y$ . Based on these variables, the plug-in determines the corresponding minimum and maximum values as seen in the log, e.g.  $x_{\min} = 0, x_{\max} = 100, y_{\min} = 1000, y_{\max} = 10000$ . For map of size, for example,  $800 \times 600$  pixels (i.e.,  $w = 800$  and  $h = 600$ ), the plug-in thus specifies  $x'_{at} = ((payout - 0)/(100 - 0)) \cdot 800$  and  $y'_{at} = ((amount - 100)/(10000 - 1000)) \cdot 600$ .

*Deadline/Timeline Map.* Activity instances are positioned along the  $x$ -axis according to the time that they have been active. When an activity instance has just become active, its  $x$ -coordinate is equal to  $w$ . Its  $y$ -coordinate is obtained by choosing a numerical variable  $v_y$ , extracting its current value, and using the same computation as for the Cartesian map. As time progresses, the  $x$ -coordinate changes (becomes less and less), but the  $y$  coordinate remains constant. The position function for a time map  $l$  for instances of an activity type  $ai = (at, cid)$  is defined as  $position_l(at) = (x''_{at}, y''_{at})$  where

$$x''_{at} = \left(1 - \frac{t - \mathcal{T}}{d_{at}}\right) \cdot w \quad \text{and} \quad y''_{at} = \frac{v_y - y_{\min}}{y_{\max} - y_{\min}} \cdot h$$

where  $d_{at}$  is a constant which defines the maximum valid duration for instances of an activity type  $at$ . If a certain activity instance is active for more than  $d_{at}$ ,  $x''_{at}$  will become negative and is then enumerated in the list of activity instances with invalid coordinates.

*Process-Model Map.* As mentioned before, for a process-model map activity instances are projected onto the corresponding icon in the model. In order to enable the automatic generation of maps of this type, end users need to provide a process model which also encodes the actual coordinates of the icons that represent the model's activities. Currently, we only support the Petri net formalism, though we believe that it is relatively easy to extend the implementation to support other process modelling formalisms. Petri Nets are stored in files in PNML standard format which also encodes the coordinates of positions of all activities (i.e. Petri net transitions) of the model.

## 5 A CASE STUDY WITH AN AUSTRALIAN INSURANCE COMPANY

We applied the visualisation framework in a case study that we conducted with one of the largest insurance companies in Australia, namely Suncorp, during the second-half of 2012. Through regular meetings (almost weekly) with the stakeholders from Suncorp, we identified the need to communicate the current landscape of Suncorp's claims processing performance to higher managers within the company. To this end, we believed that the visualisation framework proposed in this paper could be used to generate a number of movies summarising Suncorp's claims processing trends and performance.

Suncorp provided us with data related to the processing of claims that were finalised within a 6-month period (regardless of the starting time of the claims). The data consists of over one million events for 34 activity types, which together describe the processing of over 32,000 claims from multiple departments within Suncorp.

For the purpose of evaluating the visualisation framework and the usefulness of the resulting movies, a subset of the data (from one department only) was used. This subset of data was selected because it contains rich attribute information, including: loss type (i.e. the cause of a loss that triggered an insurance claim, such as fire, theft, or burglary), payout amount (i.e. the amount of money paid to a customer as a result of an awarded insurance claim), team (i.e. the team within Suncorp which processed the claims), and many others. As will be detailed later, the richness of attribute information in this subset of the log allowed us to produce interesting maps and, consequently, movies.

### 5.1 Overview of the Maps (and Movies) Created for the Case Study

Using this subset of data, four movies were produced using four different maps. A thorough explanation of the interpretation of these maps and the various interactive configuration options available to users (while a movie is being played) is provided in the remainder of this section. Some screencasts of the tool showing these four movies are available at <http://www.processmining.org/online/logonmaps>.

#### 5.1.1 Australia Map

The first movie was produced using a geographical map of Australia. In this movie, a dot is projected onto the map at the position that corresponds to the state or territory where the claim was lodged. Thus, the goal of this map is to display the distribution of insurance claims across all Australian states and territories at any given point in time, and how the distribution evolves over a period of time. Figure 1 shows a snapshot of the movie.

The bottom part of Figure 1 shows the widgets to control the playback of the movie. From left to right, this area contains the play button to start/stop the movie, the timeline and time slider box which shows the relative progression of the movie and the slider that can be

dragged by users to reach a particular snapshot, the clock showing the current timestamp of the movie being played as obtained from the timestamp information in the event log, and the slider to adjust the playback speed on the fly. As also stated in [24], it is important to play the movie at the right speed. The optimal speed may be hard to predict: animations played too slowly may become boring, whereas the opposite can cause relevant information to be missed. The timeline box also shows a wave in which the  $x$ -axis represents time and the  $y$ -axis the number of active activity instances. The purpose of the two buttons above the timeline box (labelled as 'Pos. Trend' and 'Colour Trend') will be explained later in this section.

At the top of Figure 1, we see the Maps panel which is made up of a number of tabs, each associated with a different movie. By selecting a tab, the corresponding movie is brought to the front to show the state of the process at a 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 animation they are watching in case they wish to consider a different type of movie. In Figure 1, two animation movies were executed simultaneously: the first one is the Australian map animation (called the 'Loss Cause/the State' plot on the tab) and the other one is the 'Incurred Amount/Claim Duration' movie (not shown in Figure 1, but users can switch between the two movies at anytime).

There are two boxes on the panel on the left-hand side of the screenshot. The top box enumerates the activity instances for which there are no associated positions (i.e. they do not belong to the domain of the corresponding position function). The bottom box enumerates those activity instances whose positions are 'invalid' (i.e. the corresponding position function returns coordinates which either fall outside the map boundary or could not be evaluated). In the screenshot shown in Figure 1, there was no activity instance that was projected as 'invalid' at that particular point in time.

The centre of Figure 1 shows a snapshot of the movie being played. The configuration file used to produce this movie projects activity instances onto the corresponding Australian state/territory in which the claim was lodged. The size of the dots is positively correlated with the number of events that are simultaneously present on the map at exactly the same position. As mentioned previously, dots projected at the same coordinates are merged to form bigger dots; moreover, the colouring scheme of the dots is configurable. In Figure 1, the colour of the dot was configured so that each dot is coloured according to the 'loss cause' of the insurance claims. The association of colours with values of loss cause is explained in the legend shown on the right-hand side panel. Many dots are projected at the same coordinates and, hence, they are merged in clusters. As mentioned previously, each bigger dot is divided in as many slices as the number of different colours associated with merged dots. The size of the slice is determined

proportionally with the percentage of dots of a given colour: the number of activity instances with a particular colour (i.e. a particular loss cause) in a dot over the total number of activity instances represented by the dot. These percentages can be viewed by rolling the mouse over the dots (as shown in Figure 1).

On the right-hand side of Figure 1, there are two tabs. The top one is labelled as 'Parameters' and the bottom one is labelled as 'Legend' (the 'Legend' tab is not shown in Figure 1 because the tab was being selected in the screenshot, causing the 'Legend' tab itself to disappear). As explained above, the purpose of the 'Legend' tab is to describe the meaning of each colour in the dot. The 'Parameters' tab, on the other hand, is used to configure the colouring scheme and the manner in which the dots are to be displayed in the animation movie.

Figure 3 (top part) shows the expanded parameter tab. The first check-box allows a user to merge multiple dots into one dot even if the dots are partially overlapping. If this option is ticked, dots whose area partly overlap are merged. Otherwise, if it is unticked, dots are merged only if they are exactly positioned at the same coordinates. The second check-box allows the fading out of the dots when the corresponding activity instances are soon going to be no more active. The number of frames required for the dots to begin to fade out can be customised using the slider on the panel. When dots are merged, by default, the bigger dots are annotated with the number of dots that are merged. Sometimes the process owner does not want to disclosure this aggregate information for privacy and/or confidentiality reasons. For instance, this happened for the Suncorp's case studies. The last check-box allows users to display or remove the exact number of merged dots that is shown in the middle of each dot: If this option is unticked, the number will not be displayed in the movie (the legend will also show an 'N.A.' status in place of the exact number).

The second half of the parameter panel allows users to customise the colouring scheme of the dots in the movie according to one of the schemes described in Section 3. The three options will respectively enable the colouring of dots based on the state, the age, or the values of a particular variable of the corresponding activity instances. Furthermore, for the third option, the variable type to be used is selected from a drop-down box, as shown in Figure 3.

The 'Pos. Trend' panel, shown in Figure 3, is used to present to viewers the evolution of the number of activity instances that are correctly projected onto the map (blue-lined graph) and not projected onto the map (green-lined graph) over the time span of the movie. The red-lined graph in this figure shows those activity instances that could not be projected onto the map correctly (i.e. those activity instances with 'invalid' projection); however, we do not see the red-lined graph in Figure 3 because there were no invalidly projected activity instances throughout the whole animation. The thick vertical red line represents the current time of the movie being played. The 'Colour Trend' panel serves a

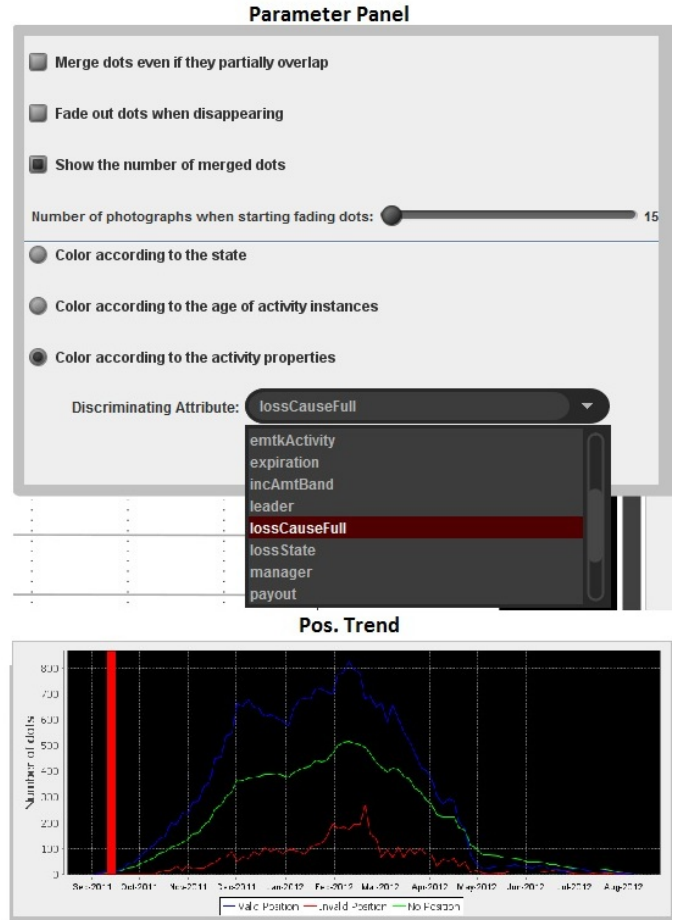


Fig. 3. Screenshots of the interactive parameter panel and the 'position trend' graph

similar function as the 'Pos. Trend' panel, except that this time it displays the evolution of the number of activity instances per dot colour over the time span of the movie.

Typical insights that can be gained from this map include an understanding of the distribution and the characteristics of claims across Australian states and, more importantly, the differences in the characteristics of the claims. For example, while playing the movie, we noticed that the state of Tasmania had a higher proportion of claims from damages of rental properties as compared to other states, while natural hazard seems to be one of the most dominant causes for insurance claims across all states.

Based on the description of our framework in Section 3, the LogOnMap plug-in is used to project individual activity instances onto the map. However, there are situations when stakeholders are more interested in the analysis of the overall distribution of cases and their characteristic, rather than of the single activity instances. This is precisely the situation that we encountered in our case study with Suncorp. In fact, two out of the four movies produced (i.e. the Australian map and the Quadrant map) are concerned with case-level performance. To address this situation, we inserted one 'dummy activity instance' for each case (i.e. trace) in the event log. These



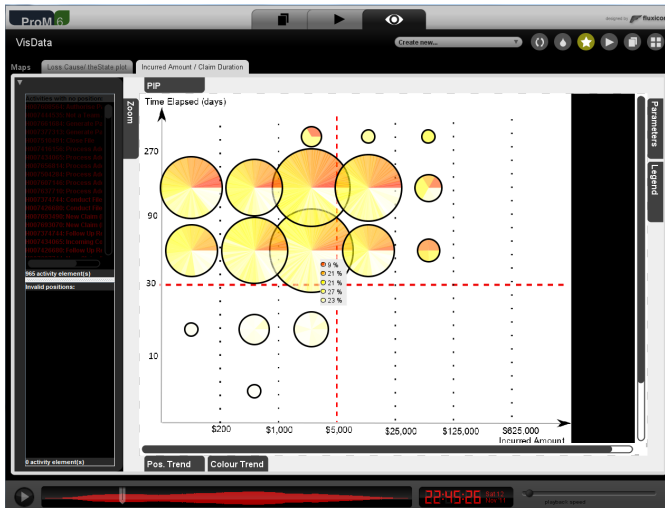


Fig. 4. A screenshot of the second animation produced using the quadrant map

‘dummy activity instances’ start before any other activity instances in the case are active and complete when all other activity instances are no longer active. Then, in the map configuration file, we project the ‘dummy activity instances’ onto the map to generate movies that show the performance and trends of cases. The other activity instances are not projected (i.e. the position function is only defined for the ‘dummy activity instances’).

### 5.1.2 Quadrant Map

The second movie was produced based on a Cartesian map whereby the  $x$ -axis represents the amount of insurance claim payouts, and the  $y$ -axis represents the number of days taken to process the claims. We call this map a ‘quadrant map’. Note that this second movie is also about viewing cases, thus the insertion of ‘dummy activity instances’ into the log was applied. A screenshot of the generated animation is shown in Figure 4.

In this screenshot, dots are coloured based on the age of the activity instances (see Section 3.2). Dots in the bottom-left quadrant represent claims with low payout values and relatively quick processing times (expected), dots in the top-right quadrant represent claims with high payout values and relatively long processing times (expected), dots in the bottom-right quadrant represent claims with high payout values and quick processing times, and finally, dots in the top-left quadrant represent claims with low payout values and long processing times (under-performing claims).

In other words, this movie allows us to gain insights into the performance of Suncorp’s claims process over a period of time. In our case study, this movie proved to be useful in conveying to business analysts and managers the performance of their claims process.

### 5.1.3 Deadline Map

The third movie was produced based on a deadline map. This movie shows those activity instances which were

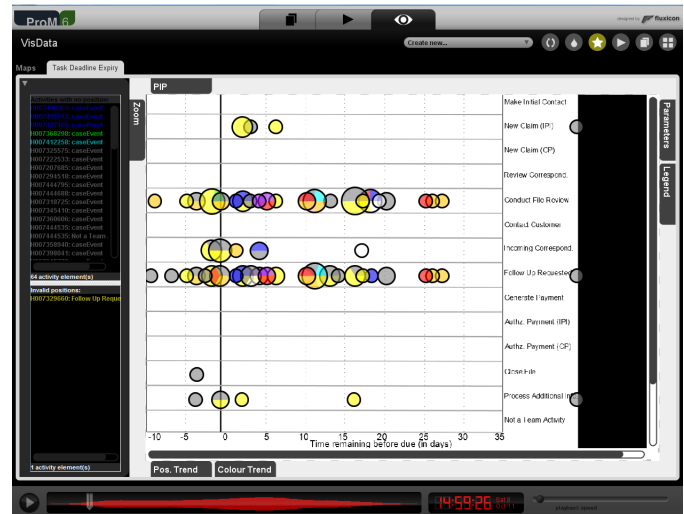


Fig. 5. A screenshot of the third animation movie produced using a deadline map

completed on-time (before the deadline) and those which were not (see Figure 5). The position of a dot on the  $x$ -axis tells us the time remaining before the corresponding activity instance reaches its deadline, while the position of that dot on the  $y$ -axis reveals the type of that instance. In this figure, dots are coloured according to the teams that performed the corresponding activity instances.

When an activity instance becomes active, a dot representing the activity instance appears on the map. The  $y$ -axis position of the dot is determined by the type of the activity instance, while the  $x$ -axis position is initially determined by the amount of time the activity instance has before the deadline is reached (note that the deadline for every activity instance is provided as an event attribute in the log). Thus, the later the deadline of an activity instance is relative to the time when the instance becomes active, the further to the right the starting position of the dot is.

After an activity instance becomes active, the dot representing that activity instance moves from right to left as time progresses. The amount of time until the deadline expires is represented through the distance to the thick black line. Consequently, at any given point in time, those activity instances which did not complete by the deadline are captured by the dots to the *left* of the thick black line.

In our case study, this movie allowed us to identify a number of activities that often ran overtime, such as the ‘Follow Up Requested’ activity, the ‘Conduct File Review’ activity, and the ‘Incoming Correspondence’ activity. Other activities, such as the ‘New Claim (IPI)’ activity, mostly completed around the deadline.

### 5.1.4 Process model map

The fourth movie was produced based on a process model map (see Figure 6). The process model used in this movie is the Fuzzy model [22] that we discovered using the Disco tool (<http://www.fluxicon.com/disco>).

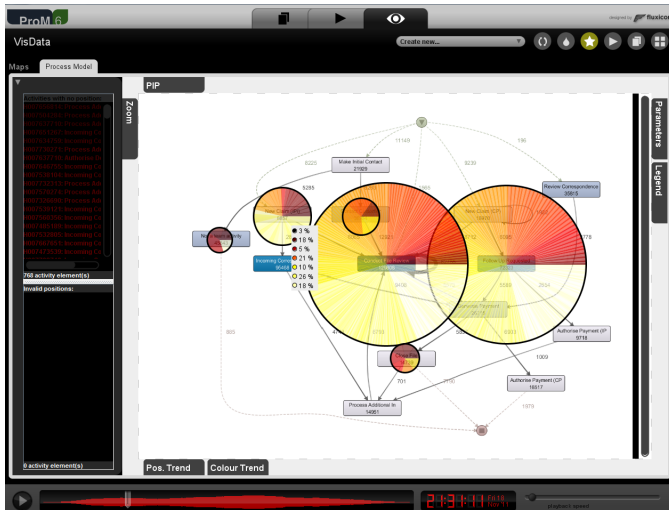


Fig. 6. A screenshot of the fourth animation movie produced using a process model map

As explained in Section 4, dots are projected onto the map according to the position of the icon representing the activity captured by the dots. In this screenshot, dots are coloured according to the age of the activity instances.

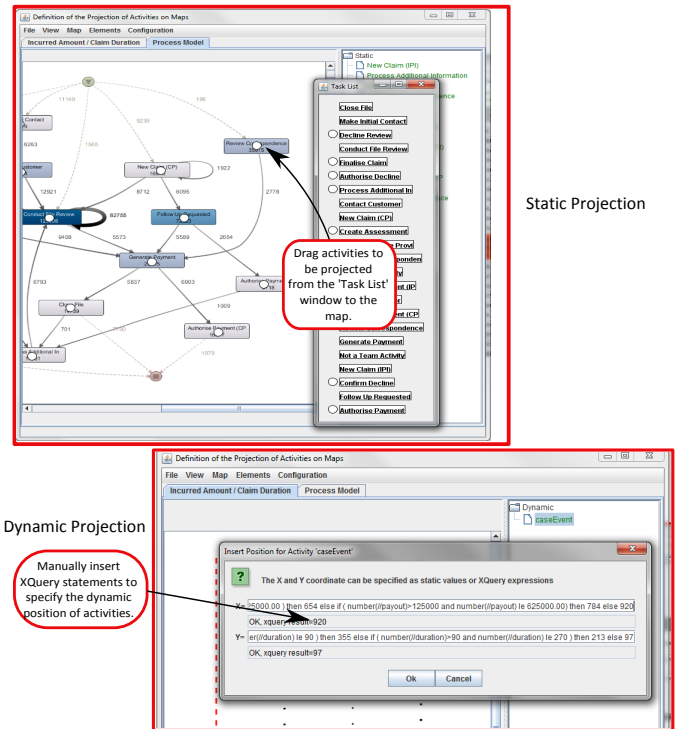
A typical insight that can be gained from using this type of map is the identification of activities in a process that can potentially be a bottleneck. For example, while playing the movie, the appearance of a large dot on a particular activity icon over an extended period of time may indicate the piling up of work items (i.e. activity instances) related to that activity. As can be seen from Figure 6, many activity instances were piling up for two activity types, namely, ‘Follow Up Requested’ and ‘Conduct File Review’.

## 5.2 Map Designer

As stated in Section 4, a *Map Designer* tool has been implemented. Figure 5.2 shows two examples of how we have used the map designer to generate the configuration files for the four movies used in the case study.

The top part of Figure 5.2 shows how we have used the map designer tool to automatically generate the configuration file for the fourth movie (the process model movie). Here, we can see how a user can simply drag an activity name (from the ‘Task List’ window) to the desired position on the map. By doing so, the map designer tool automatically generates a map configuration file which specifies, for each activity whose instances need to be projected onto the map, the *static* position of those instances.

The bottom part of Figure 5.2 shows how we used the map designer to help us generate the map configuration for the second movie (the quadrant movie). This movie requires a dynamic positioning of dots based on the variable values of the activity instances to be projected. Thus, to enable such a ‘dynamic’ positioning of dots,



Dynamic Projection

Static Projection

Fig. 7. Screenshots of the Map Designer showing static (top) and dynamic (bottom) activity projections

we inserted the desired XQuery expressions into the corresponding pop-up window.

Overall, we found the map designer tool to be quite useful in enabling us to quickly define, and adapt, our map configuration files to suit the type of visualization that we would like to see.

## 6 EVALUATION WITH END USERS

The validity, the usefulness, and the intuitiveness of the approach and of the resulting implementation has been thoroughly assessed through engagement with end users. Specifically, the approach’s validation has been conducted in two phases.

A first version of the tool was released in the second half of 2012 and reported in [8]. This version was evaluated with three subjects of a Dutch municipality: a process management specialist, a communication and marketing specialist, and a business advisor for customer contacts. In this case, we used a real-life event log concerning the process to handle the applications of the house-building permits submitted by Dutch residents. In particular, we defined four maps, as reported in [8].

Through this evaluation process, we discovered a number of usability issues and missing features in our tool which contributed to unnecessary complication in the interpretation of the results. A summary of the identified issues is provided below (see [8] for details):

- In the first version of the tool, activity instances left no trace when they disappeared. The subjects

found this particularly confusing as moving dots can disappear at different positions on the map.

- The subjects interviewed remarked that it was sometimes unclear how long activity instances were active.
- Activity instances could not be related to characteristics of the case, e.g., the type of permit requested.

After addressing the issues above (and other minor issues), we released a second version of this tool which is the version discussed in this article. For instance, the fading effect was introduced to make it more evident when activity instances are about to disappear (see end of Section 3). The different colour schemas discussed in Section 3.2 were introduced to relate activity instances to case characteristics or to draw attention to the age of activity instances.

After releasing the second version of the tool, we performed a more extensive session of experiments where users personally interacted with the tool. The participants of this second experiment session were SunCorp employees and the movies used in the experiment were those four movies already explained in Section 5. The use of a different case study from another continent allowed us to assess the framework in different settings and with subjects with a different cultural and work background.

The evaluation was conducted using an established methodology, in addition to a number of interviews with a relatively large number of subjects. Section 6.1 details the evaluation methodology, the background of the participants, and the experiment procedure. Section 6.2 reports the result of the evaluation and lessons learned, along with directions for future development.

## 6.1 Methodology for the Evaluation

The evaluation of the second version of our tool was conducted using the *Co-operative Evaluation* methodology, which is a mature, fully-documented methodology in the field of human-computer interaction [25]. This is a cost-effective technique for identifying usability problems in prototype products and processes. The technique encourages design teams and users to collaborate in order to identify usability issues and their solutions.

### 6.1.1 Nature and Number of Participants

The participants of this experiment consisted of SunCorp's employees of various roles: five team leaders, one manager, two business analysts, and one claims officer. They had different levels of knowledge of the insurance claim process.

In terms of the participants familiarity with process mining techniques and business process management technology, three out of the nine participants were aware of the existence of Business Process Management systems, while the rest were not aware at all, or only had a very limited awareness, of such systems. Furthermore, only one participant had experience with process analysis, while the rest had none, or limited, experience with process analysis.

It is worth highlighting that, because of the user-intensive nature of this method, it is difficult to run this experiment with a large number of users. Nevertheless, as documented in [25], past applications of such a method have shown that the careful choice of experiment subjects, even if relatively small, can minimise the problem of obtaining subjective results.

### 6.1.2 Procedure to Conduct Experiments

The experiment of our visualisation tool was conducted using the four movies generated using the *LogOnMapReplay* plug-in (detailed in Section 5). The experiment was conducted with each participant individually, one after another. Before the experiment started, we gave a brief introduction of the framework and of the four movies, after which we let the participant play with the tool on their own. Each participant was roughly given 10 minutes to interact with the tool. In accordance to the methodology, no further comments were given, thus letting the participants draw their own conclusions. Without our interference, we could thus evaluate the level of understandability of the map metaphor and the usefulness of the approach when extracting knowledge from event logs.

While performing such tasks, participants had to explain what they were doing by 'thinking-aloud'. During the experiment, notes were taken on the behaviours exhibited by the participants in order to measure the degree of efficiency and effectiveness observed. In particular, they were asked to communicate any meaningful conclusions that they managed to draw by observing the animations and interacting with the tools.

At the end of the experiment, each subject was given the opportunity to fill out a semi-structured questionnaire with questions designed to measure the subjects' impressions and expectations with regards to the tool. To ensure anonymity, the filled-out questionnaire was inserted into a ballot box by the participant him/herself, thus guaranteeing the anonymity of responses.

In summary, our experiment methodology provides a valuable means to not only verify the effectiveness and efficiency of the visualisation framework, but also to elicit further possible improvement opportunities (see Section 6.3). This method is, therefore, an eminently formative evaluation method, rather than a summative one. It is useful for identifying those usability 'bugs' that can affect the effectiveness of the system being evaluated.

## 6.2 Evaluation Results

The results of our experiment will be detailed based on the questionnaire results (both quantitative and qualitative data were collected) and our observations of the participants' responses to the tool during the experiment. The first two questions in the questionnaire (Q1 and Q2) were used to gauge participants' familiarity with BPM systems and process analysis (the result of which has been summarised in Section 6.1.1). The rest of the questions (Q3–Q8) were used to gauge participants' satisfaction with the tool.

### 6.2.1 Questionnaire results

The questionnaire used consisted of both closed and open questions. The third question (Q3) asked participants the type of insights that they expected to obtain by using the visualisation tool. Participants' expectations of insights they wish to obtain by using the tool vary, although they are roughly consistent with what can be expected from visual analytics. A baseline expectation from all participants is to understand the *trend* and *volume* of their claims processing in terms of performance, and to understand *why* certain trends occurred. Some participants also expected the tool to guide them in identifying *problems* with their processes and how to resolve them. Finally, a few participants also expected the tool to help them identify *opportunities* in their processes with a focus on planning the assignment of resources.

The remaining five questions (i.e. Q4–Q8) consisted of both closed and open-ended questions. For each of these questions, the user selects one out of a number of pre-determined satisfaction rating (e.g. 'very much', 'much', 'not so much', 'not at all', and 'I don't know'). Furthermore, to gain more insights, they were allowed to provide reasons for the satisfaction rating they selected.

The results of the closed questions are displayed in Figure 8. With respect to Q4, most participants (8 out of 9) found the tool allow them to gain expected process-related insights (expressed as answer to Q3). This is confirmed by analysing the related comments, e.g. *"Having a visual representation makes it clearer [...] the flow on impacts when there is a problem in one area and how it then relates to another area."*, *"Was good to see key area for the business to improve on [...]."*, *"You could clearly identify bottlenecks, claims with long duration but low value etc."*, *"Give graphical representation of claims incidents and work loading. Give insights to claims costs compared to time of year."*

However, there was one participant who did not find the tool to help him/her much with addressing the question he/she listed earlier. Upon reading the comment, it turned out that the participant expected a feature in our tool which was simply not built: the ability to 'drill down' into activity instances that appear on the movies to learn more about them (e.g. obtaining the details of the customer related to a particular activity instance).

Another recurrent comment was about the intuitiveness of the deadline map, with dots moving towards the left as time progresses. They would have expected them to move towards the right. This is an interesting point: in the first version of the tool, the dots moved towards the right. We changed this since the subjects from the Dutch municipality found the movements towards right as not very intuitive. This makes us suspect that the intuitiveness may also be subjective and depend on the cultural background.

In terms of the maturity of the tool (Q5), it was quite surprising to note that one third of the participants did not provide any responses; however, by looking at the related comments, it turned out that these participants did not respond because they either did not understand the question or they found that they had not spent enough

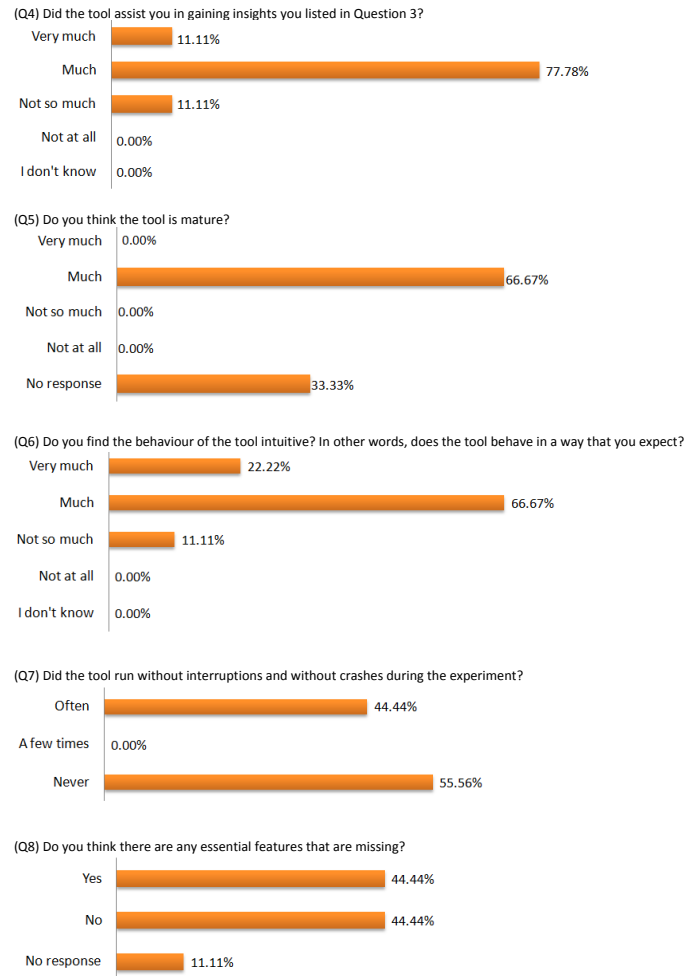


Fig. 8. Closed questions results

time with the tool to properly answer Q5. Nevertheless, those who responded to this question all agreed that the tool is mature. The related comments also confirmed this observation, e.g. *"...it captures a lot of relevant information and provides plenty of options to navigate to specific areas you might want to examine..."*, *"...it has all the activities for 6 months and I can go back in time if I wanted to. Obviously as we used it we would discover more things we may want to see but for the moment, I think there is enough for me to play with."* At the same time, comments related to Q5 also suggested a number of possible tool improvements, including the use of 'plain English' in the description and labelling of various configuration options.

Q6 evaluates the intuitiveness of the tool. As shown in Fig. 8, the responses we obtained were quite similar, with 8 out of 9 participants finding the tool quite intuitive. From those who found the tool to be intuitive, we received comments such as *"It acts exactly to what you would expect, according to the selections you choose."*, *"Visual interpretation easy to accept."*, and *"[The] tool is easy to use."* The one participant who did not find the tool to be intuitive commented on the need for the tool to have clear labels, filters, and 'help' boxes so that *"we can easily navigate around it."*

Another feature of the tool that we wanted to evaluate was its operational stability. Therefore, Q7 asked participants to rate whether the tool ran smoothly during the experiment. The responses to this question are almost evenly divided, with 5 participants stating the tool to be running smoothly during the experiment and another 4 participants stating the opposite. One participant commented that it was frustrating that he/she managed to crash the tool, and other participants commented on the low-quality graphics used during the experiment due to the projector used.

Finally, we also asked participants if there were any features in the tool that they would like to have but that were not currently available. Apart from one participant who did not respond, the responses to this question are evenly split, with 4 participants stating that there were missing features and another 4 participants stating the opposite. Among those who responded ‘Yes’ to this question, a number of features that should be added were suggested. These suggested features, except for those that have already been stated in response to Q6, will be detailed in Section 6.3.

### 6.2.2 Additional Observations

As explained in Section 6.1, we took notes to try and gauge the degree of efficiency and effectiveness as experienced by the involved subjects during the empirical tests. In particular, we were interested to learn whether the subjects could draw interesting conclusions, which were either unexpected or confirmed previous intuitions.

The notes taken during the experiment stated that participants managed to derive important conclusions as suggested by the movies. For example, one participant gained insights with regards to the distribution of claim types across different Australian states and raised the possibility of how insurance premiums can be adjusted accordingly. A number of other participants gained insights into the ‘absurdity’ of having small-value claims which took a very long time to complete. One participant clearly acknowledged the usefulness of the deadline map to compare performance levels across different teams. Another participant noted a peak time period in terms of the number of claims. Overall, the insights gained by participants from using this tool are consistent with what we expected the tool to be able to provide.

The rest of the observation notes taken during the experiment corroborated the results of the questionnaire, thus further validating the results of our experiment.

## 6.3 Evaluation Conclusion and Future Tool Improvement

The analysis of the questionnaire’s answers and of the users’ interaction with the tool let us make the following observations:

- 1) In most cases, the tool behaved as expected.
- 2) A number of participants highlighted that the language used in the tool was not written in ‘plain

English’, leading to potential wrong interpretation of the options or features.

- 3) The stability of the tool needed to be improved as it crashed rather frequently during the experiment.
- 4) A recurrent request from many participants was to provide filtering capabilities such that the movie can be configured to display only information of interest, e.g. only display activity instances which were executed by a certain group of resources or which occurred within a specific time period. Note that ProM already has sophisticated log filtering capabilities; nevertheless, it is still worthwhile to integrate them directly into this plug-in to allow users to do the filtering on the fly while playing the movie.
- 5) One participant expressed his/her interest in being able to *drill down* into the activity instances associated with certain colours or having certain characteristics, and to extract the respective details.
- 6) For dots representing more than one activity instance, participants showed an interest in being able to quickly learn the percentage of slices of each represented colour (e.g. 30% of slices are coloured red, 10% blue, etc.). In the version evaluated, we simply showed the number of dots that were amalgamated, without detailing this statistics per colour. Since then, this issue has been addressed. Moreover, more detailed bug testing has been conducted and, as result, many bugs have been fixed.

Despite the issues listed above, we believe that the evaluation has clearly assessed the validity of the tool. The concept and the design of the tool are desirable. The metaphor of maps and movies is clearly understood and serves the purpose of gaining insights into past executions to understand process trends and performance.

Overall, subjects were enthusiastic about the ability to generate movies based on event data. Several participants expressed a desire to start using the plug-in in their own analyses. In fact, the tool has helped some end users find interesting patterns in various situations, some of which were quite surprising to them. Therefore, we conclude that the concept of our visualization framework is promising and that the current tool already illustrates its potential.

## 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 visualise 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, bottlenecks, etc.

This paper describes an implementation of these ideas in the ProM framework. We also developed a map designer that allows end users to define the maps of interests and to position activities on them. Many maps are widely-applicable (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 also developed a ProM plug-in that semi-automates the definition of certain types of maps, by self-generating the map picture and the projection of activity instances. 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 various schemas according to the properties of the corresponding activity or process instance.

Interested readers can try the ProM implementation with a sample event log. The sample event log, the corresponding maps, the configuration files, and the related instructions are available from <http://www.processmining.org/online/logonmaps>.

The approach has initially been evaluated using a case study in the context a Dutch municipality. The outcome triggered a series of changes in the framework and the reference implementation. After the adjustments, we performed a second more extensive experiment where participants actually worked with the tool. To minimize the issue of obtaining subjective results, this experiment was conducted based on a well-founded methodology, i.e. the *Co-operative Evaluation* methodology [25]. This second experiment was conducted with the employees of Suncorp (one of the largest insurance organisations in Australia).

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 visualisations such as the “process movies” proposed in this paper.

## REFERENCES

- [1] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, “Big data: The next frontier for innovation, competition, and productivity,” McKinsey Global Institute (MGI), Tech. Rep., May 2011.
- [2] D. Laney, A. Bitterer, R. Sallam, and L. Kart. (2012, December) Predicts 2013: Information innovation. Gartner.
- [3] W. M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Berlin Heidelberg: Springer-Verlag, 2011.
- [4] J. J. Thomas and K. A. Cook, Eds., *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE CS Press, 2005.
- [5] D. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, Eds., *Mastering the Information Age: Solving Problems with Visual Analytics*. VisMaster, <http://www.vismaster.eu/book/>, 2010.
- [6] M. de Leoni, M. Adams, W. M. P. van der Aalst, and A. H. M. ter Hofstede, “Visual support for work assignment in process-aware information systems: Framework formalisation and implementation,” *Decision Support Systems*, vol. 54, no. 1, pp. 345–361, 2012.
- [7] W. M. P. van der Aalst, M. de Leoni, and A. H. M. ter Hofstede, *Computational Intelligence*. Nova Publisher, 2012, ch. 8: Process Mining and Visual Analytics: Breathing Life into Business Process Models, pp. 107–138.
- [8] M. de Leoni, J. Buijs, W. M. P. van der Aalst, and A. H. M. ter Hofstede, “Facilitating process analysis through visualising process history: Experiences with a dutch municipality,” Eindhoven University of Technology, Tech. Rep. BPM-12-24, 2012.
- [9] C. Chen, *Information Visualization: Beyond the Horizon*. Springer-Verlag, New York, Inc., 2006.
- [10] R. Spence, *Information Visualization: Design for Interaction*, 2nd ed. Harlow, England: Pearson Education Limited, 2006.
- [11] G. Alonso and C. Hagen, “Geo-Opera: Workflow Concepts for Spatial Processes,” in *SSD’97: Proceedings of the 5th International Symposium on Advances in Spatial Databases*, ser. Lecture Notes in Computer Science, vol. 1262. Springer Verlag, 1997, pp. 238–258.
- [12] D. Kaster, C. Bauzer-Medeiros, and H. V. da Rocha, “Supporting Modeling and Problem Solving from Precedent Experiences: The Role of Workflows and Case-based Reasoning,” *Environmental Modelling and Software*, vol. 20, no. 6, pp. 689–704, 2005.
- [13] B. Schönhage and A. Eliëns, “Management Through Vision: A Case Study Towards Requirements of BizViz,” in *AVI 2000: International Conference of Information Visualisation*. IEEE Computer Society, 2000, pp. 387–392.
- [14] D. Oelke, C. Ming, C. Rohrdantz, D. Keim, U. Dayal, H. Lars-Erik, and H. Janetzko, “Visual Opinion Analysis of Customer Feedback Data,” in *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (IEEE VAST 2009)*. IEEE, 2009, pp. 187–194.
- [15] D. Keim, T. Nietzsche, N. Schelwies, J. Schneidewind, T. Schreck, and H. Ziegler, “A Spectral Visualization System for Analyzing Financial Time Series Data,” in *Proceedings of the Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis 2006)*. Eurographics Association, 2006, pp. 195–200.
- [16] H. Ziegler, T. Nietzsche, and D. Keim, “Relevance Driven Visualization of Financial Performance Measures,” in *Proceedings of the Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis 2007)*. Eurographics Association, 2007, pp. 19–26.
- [17] P. Bak, F. Mansmann, H. Janetzko, and D. Keim, “Spatio-temporal Analysis of Sensor Logs using Growth Ring Maps,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 913–920, 2009.
- [18] D. Keim, “Visual Exploration of Large Data Sets,” *Communications of the ACM*, vol. 44, pp. 38–44, 2001.
- [19] M. Krstajic, E. Bertini, and D. A. Keim, “Cloudlines: Compact display of event episodes in multiple time-series,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2432–2439, December 2011.
- [20] J. Kehrer and H. Hauser, “Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 3, pp. 495–513, 2013.
- [21] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski, “Visualizing time-oriented data—a systematic view,” *Journal on Computers and Graphics*, vol. 31, no. 3, pp. 401–409, 2007.
- [22] C. W. Günther and W. M. P. van der Aalst, “Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics,” in *International Conference on Business Process Management (BPM 2007)*, ser. LNCS, vol. 4714. Springer-Verlag, 2007, pp. 328–343.
- [23] J. Li, J.-B. Martens, and J. J. van Wijk, “A model of symbol size discrimination in scatterplots,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’10)*. ACM, 2010, pp. 2553–2562.
- [24] J. Heer and G. Robertson, “Animated transitions in statistical data graphics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1240–1247, November 2007.
- [25] A. Dix, J. E. Finlay, G. D. Abowd, and R. Beale, *Human-Computer Interaction*, 3rd ed. Prentice Hall, 2003.