Noname manuscript No.
(will be inserted by the editor)

Automatic linguistic reporting of customer activity patterns in open malls

Manuel Ocaña*† · David Chapela-Campa§ · Pedro Álvarez‡ · Noelia Hernández† · Manuel Mucientes§ · Javier Fabra‡ · Ángel Llamazares† · Manuel Lama§ · Pedro A. Revenga† · Alberto Bugarín§ · Miguel A. García-Garrido† · Jose M. Alonso§

Received: date / Accepted: date

Abstract In this work, we present a complete system to produce an automatic linguistic reporting about the customer activity patterns inside open malls, a mixed distribution of classical malls joined with the shops on the street. These reports can assist to design marketing campaigns by means of identifying the best places to catch the attention of customers. Activity patterns are estimated with process mining techniques and the key information of localization. Localization is obtained with a parallelized solution based on WiFi fingerprint system to speed up the solution. In agreement with the best practices for human evaluation of natural language generation systems, the linguistic quality of the generated report was evaluated by 41 experts who filled in an online questionnaire. Results are encouraging, since the average global score of the linguistic quality dimension is 6.17 (0.76 of standard deviation) in a 7-point Likert scale. This expresses a high degree of satisfaction of the generated reports and validates the adequacy of automatic natural language textual reports as a complementary tool to process model visualization.

1 Introduction

SOcial Workflows (SOW) coordinate the activities carried out by a group of users [22] who either individually or in cooperation try to achieve a certain objective. The SOW are unstructured flows in which a large number of users carry out activities of a very diverse nature that they spread over time and typically consume few computing resources. An example of processes that

†Universidad de Alcalá, Alcalá de Henares, Madrid, Spain,
E-mail: *mocana@depeca.uah.es ·
‡University of Zaragoza, Zaragoza, Spain ·
§Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Santiago de Compostela, Spain

are modeled through these types of workflows are the marketing campaigns that have the objective to motivate and persuade potential customers in the consumption of a certain product or service. However, from a technological point of view, the characteristics of the SOW are: (i) they model unstructured processes where users have multiple options and/or paths, and where they can even carry out additional activities to those initially planned in the workflow; (ii) a large number of users; and (iii) depending on the skill or motivation of each user, the same activity can be performed immediately or have a temporary duration much higher (in the order of hours or even days).

These characteristics add a series of challenges when managing and monitoring the execution of SOW. On the one hand, the management of this type of workflow requires the handling of a large number of simultaneous executions whose duration extends over time. On the other hand, the analysis of the execution of the workflow requires the use of process mining techniques [6] to obtain the real workflow followed by the users and, in this way, understanding what really happens in the execution of the process, thereby allowing the improvement of the process and its (dynamic) adaptation to the needs of users. However, for unstructured processes with a huge number of executions, the results of the current process discovery algorithms are workflows that have a structure in spaghetti and therefore very difficult, if not impossible, to interpret. Therefore, it is necessary to develop techniques to extract valuable information about the discovered workflow. Furthermore, this information should be conveyed to the users who will consume it in an understandable way. To this aim, we combine in our reports both graphical information about the process as well as automatically generated linguistic reports which summarize the relevant information about the processes. This is the first axis around which pivots the project Business (Artificial) Intelligence for Social Workflows (BAI4SOW).

Furthermore, the BAI4SOW project adds another dimension to the SOW: the activities to be carried out by users take place in geographic locations (e.g., an open shopping center, an area of tourist interest, an area where some activity is carried out or, in general, any area of a city) and they are also geolocated activities that users carry out through mobile devices (e.g., activities in a marketing campaign). Therefore, it is necessary to capture the events related to the execution of this type of activities and, for this, it is required the development of localization techniques and detection of the behavior of users in a given geographic area. This is the second axis of the project BAI4SOW.

In addition, given the high number of users who can potentially participate in this type of flow, is necessary to run the algorithms in grid computing, cluster, and cloud infrastructures, minimizing the cost that this computation could have in business clouds like Amazon and Google. This constitutes the third axis of the BAI4SOW project.

In this article, we present the complete system implemented in the project BAI4SOW, which enhances the proposal made in [33]. This system consists of a localization system to locate the customers and identify the activities that they are carrying out, and of the application of process mining and natural language generation techniques to this information in order to analyze the behavior of the customers to extract insights and linguistic reports from it. We present the results obtained in BAI4SOW project, whose objective is in the development of both process mining algorithms for the analysis of SOW containing geolocated activities, and automatic linguistic reports that can be helpful for marketing campaign design.

The rest of this manuscript is organized as follows. Section 2 presents the main structure of the BAI4SOW project and its framework. Then, the three main parts of the presented framework are described: Section 3 introduces the localization system and its parallelization; Section 4 gives an overview of process mining and its application to this project; and Section 5 describes the followed natural language generation approach. Finally, Section 7 presents the conclusions of the publication, and sketches some insights for future work.

2 BAI4SOW project

The main objective of the BAI4SOW project is the development of intelligent techniques for automatic extraction and analysis of user behavior in processes modeled through social workflows. In particular, the target processes are those involving users moving around an outdoor area and carrying out the activities of the social workflow through their mobile devices. Fig. 1 depicts the conceptual framework which is developed to achieve this objective. It includes the following components:

– *Localization and behavior detection algorithms.* To analyze social workflows is necessary to record the events generated by the activities that each user carries out. For each instance of the process —an execution of the process from start to end—, a sequence composed of these events is generated. In those processes involving users moving around a geographic area, this activities' model when a user arrives at a certain point (to, for instance, make a purchase or take a photo), sits and moves inside an establishment, etc. Therefore, it is mandatory to *locate* the users in the geographic area where they are moving, as well as to detect users' behavior of interest once they are in a given location. To record the execution of these activities, a set of algorithms have been developed using the information given by the users' mobile devices such as the Global Positioning System (GPS) and Wi-Fi signals sensors.
– *Parallelization techniques.* These techniques are used to design a technological infrastructure based on the integration of *cloud* resources that support the efficient and large-scale processing of the algorithms developed during the project. This infrastructure facilitates making a decision on what computation and storage resources are more appropriate for the execution of the algorithms.
– *Process Mining Algorithms.* This set of algorithms allow to automatically discover the workflow —also referred to as process model— that has been actually executed by the users and, also, to extract relevant information

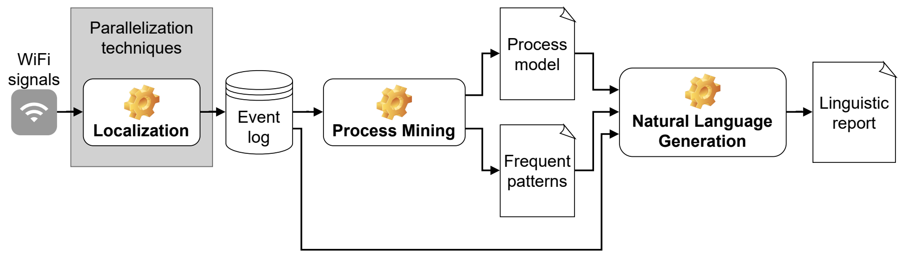


Fig. 1 Conceptual framework of BAI4SOW project.

about the process such as: *i) frequent activity patterns*, which indicate patterns of frequent user behavior; and *ii)* the *hierarchical organization* of unstructured and spaghetti workflows, which ease the visualization of the process reducing its complexity at different levels of abstraction.

– *Natural Language Generation Techniques.* This set of techniques take as input the workflows and frequent activity patterns, as long as other analytics from the recorded behavior, and automatically generate textual descriptions providing textual information in natural language about their most relevant characteristics. This is an additional tool for conveying information to users, complementary to the visualization of workflows.

The computational solution proposed in this project, which is depicted in Figure 1, is mainly deployed in a cloud-based architecture. The mobile devices record the WiFi signals from the mall and send the data to the cloud-based localization system. This system processes the data and produces the event log with the activities that each customer carried out. The process mining and natural language generation algorithms, which are also deployed in a cloud system, access the event log to work with it. Process mining techniques produce the process model and the frequent activity patterns, and then the natural language generation techniques create the linguistic report. All these outputs are generated in the cloud to be accessible by an analyst from a client application, in order to analyze the behavior of the customers.

It is worth noting that this conceptual framework is applicable to SOW containing activities related to the location and behavior of the users taking part in them. However, process mining algorithms and parallelization techniques are directly applicable to any workflow in general, and particularly to unstructured processes with many users taking part in them.

In the following sections we will describe the three main blocks of this analysis: *i)* the implemented localization system, and its parallelization, to obtain the activities that the users are carrying out; *ii)* the process mining techniques used to discover the behavior of the customers; and *iii)* the natural language generation approaches applied to automatically generate linguistic reports.

3 Localization system

In this project, we consider the customers' behavior in the mall as a process, and the actions they carry out as the activities of the process. In this context, an execution of the process is the sequence of activities that a customer carries out in the mall, from the entrance to the end. For this reason, in order to analyze the behavior of the customers, it is mandatory to register the activities that they carry out while in the mall. This information is stored in what is called an *event log*, which constitutes the foundation of process mining analyses (as we will introduce in Section 4). For this reason, the process to locate the customers and identify what are they doing is crucial, so it must be ubiquitous, accurate, and in real-time. This section describes the localization system developed in this project and the parallelization techniques applied to increase its efficiency.

3.1 Indoor WiFi-based system

Outdoor localization is usually obtained through the use of GPS but, unfortunately, it is not suitable for indoor environments – such as the environment of this analysis – due to the Non-Line-Of-Sight (NLOS) effect. To solve this issue, the proposed method uses a fingerprint WiFi-based localization as a highly-available, no-installation, no-cost solution: there are WiFi Access Points (APs) in almost every shop, almost every smartphone has a WiFi interface and there is no need to be connected to WiFi networks to measure WiFi signal, so it is free of charge even for private networks.

This kind of methods relies on a two-stage approach: (1) the training stage, when the environment is divided into cells where real WiFi measurements are collected and stored to build the localization system; and (2) the localization stage, when new measurements collected at an unknown location are compared with the stored ones to estimate the location of the device.

Our localization system is based on the work proposed in [24]. It implements a method to increase the localization resolution without the need of increasing the number of cells to site-survey at the cost of a higher computational cost. Of course, real-time is a requirement for our system, so we have implemented a new version using parallel techniques (as will be explained in Section 3.2).

Fig. 2 shows the general architecture of the system. As it can be seen, during the training stage a Support Vector Regresion (SVR) algorithm is used to extrapolate virtual WiFi measurements at positions where no real samples were collected using the real collected information from each existing WiFi AP. The trained localization system will be composed of all these surfaces.

Then, during the localization stage, customers will collect new measurements in real-time at their unknown locations. These new measurements will be compared with the values stored in the corresponding surfaces to find their location inside the environment. This way, the coordinates where the differ-

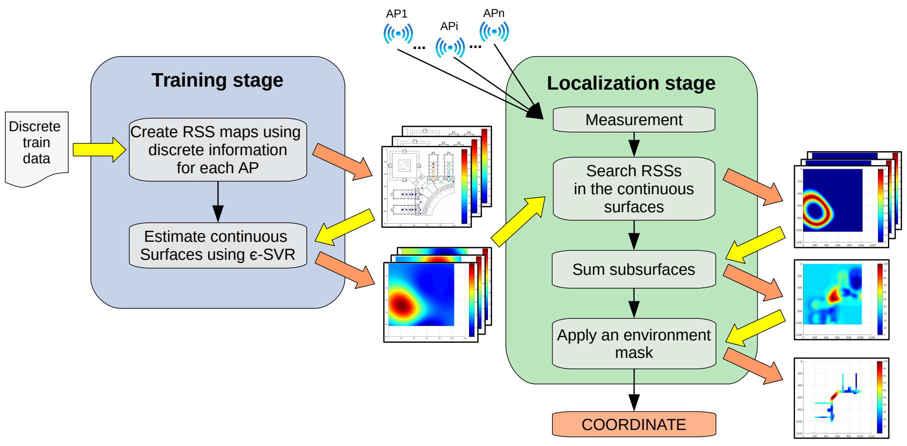

AP1
APi
APn
Training stage
Discrete train data
Create RSS maps using discrete information for each AP
Estimate continuous Surfaces using ϵ-SVR
Localization stage
Measurement
Search RSSs in the continuous surfaces
Sum subsurfaces
Apply an environment mask
COORDINATE

Fig. 2 General architecture of the WiFi-based indoor localization system [24].

ence between the Received Signal Strength (RSS) from an AP and the stored RSS for that coordinate is low will be marked with higher scores than the coordinates with greater differences. Finally, the scores for each coordinate will be summed up, to obtain a resulting surface where the coordinate with the highest score is the most likely location of the customer.

In [24], this localization method was evaluated for determining the optimal size of cells when it is applied in a medium-size indoor scenario (around 2.500 m^2). The results proved that 15 cm side cells are an efficient option using all the detected APs. With this configuration, and the size of the open mall (see Section 6), the execution time is 44.4 hours to compute all the surfaces during the training stage and 325 seconds to determine the current location of a customer. The experiments were conducted using a server with 8 Intel i7-4790K CPUs at 4.00GHz, a local SSD disk, and 32GB of RAM. Obviously, these performance results show the necessity of accelerating the training and location algorithms in order to be used in large real scenarios.

3.2 A parallel version of the location algorithms

In order to improve the performance of the previous process, new training and location algorithms have been programmed using parallelization techniques. The goal is to reduce the execution time of these algorithms in order to locate efficiently customers in a real scenario in a suitable way. The new version is based on a *master-worker architecture* [9], which has been developed to be deployed on a high-performance computing environment, specifically, in a set of cloud computing resources with the flexibility to support the dimensions and large-scale requirements of the scenario.

The master-worker architecture follows a high-level design pattern that facilitates the parallel execution of applications composed of a set of independent tasks. The pattern consists of two kinds of processes: a master, and a pool of workers. The former is responsible for assigning tasks to workers and guaranteeing that all of them are correctly completed. On the other hand, the workers execute the assigned tasks provided by the master. The correct execution of these tasks is monitored and coordinated by the master, which also checks the termination status and the logs resulting from each process. If required, the master can execute a task again or even reassign it to a different node for its completion.

The scenario of the problem depicted in this work requires a high level of flexibility when dealing with computing resources. This architectural model is highly scalable by increasing (or decreasing) the size of the pool of workers according to the execution requirements. On the other hand, the master and the workers usually interact among themselves using an asynchronous communication mechanism. Message brokers are usually used for this purpose, demonstrating their adaptability and effectiveness in this model of architectural solutions.

The parallelization of the algorithms is separated into two different stages. First, a generic master-worker architecture has been developed in Java using *RabbitMQ* as the message broker [36]. Then, the training and location algorithms have been split into a set of independent tasks that can be executed in parallel using the master-worker implementation. The process of creating a continuous reference surface as part of the training algorithm has been programmed as independent tasks that can be computed from RSS measures. This way, the generation of the different surfaces can be easily parallelized. The location algorithm requires checking the sample of the customer's unknown position against each of the reference surfaces. Each of these checks has been also developed as an independent task.

Additionally, as part of the parallelization process, the MATLAB code used to create and process the reference surfaces was refactored and programmed using Octave [21]. This decision is based on economic criteria: the use of MATLAB in different cloud computing resources requires locating a license configured for cloud for each node or deploying a MATLAB Parallel Server, which represents an additional cost that would increase as the execution requirements did as well [32].

The parallel implementation of the localization system and its deployment in the Amazon cloud computing environment is depicted in Fig. 3. As shown, the system exposes its functionality through two Web-based interfaces as REST APIs. This allows easy and independent integration with the BAI4SOW applications.

On the one hand, the *Surface Creation Service* provides a set of operations to submit training measures and compute the corresponding continuous reference surfaces. The master-worker version of the corresponding algorithm has been integrated into the logic of the service and it has been deployed into the Amazon EC2 infrastructure [8]. As will be detailed later, the flexibility of the master-worker architecture allows us to deploy the service over different configurations of virtual instances (used for the execution of workers, mainly). The results are then stored into the *Surface database*, which has been imple-

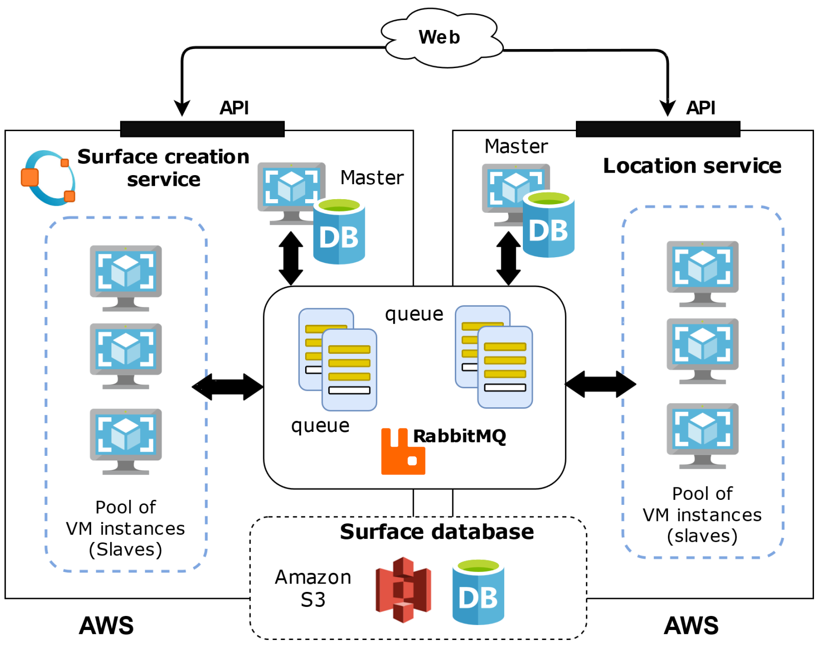


Fig. 3 Cloud-based implementation of the localization system.

mented using the *Amazon S3 service* [7]. This allows us to guarantee a robust, reliable, and durable data storage facility that can be easily scaled as well when required.

On the other hand, the *Location Service* offers the functionality to estimate users' location. The RSS samples measured by the user's mobile are aggregated and sent to the service as input. Then, this information is processed by the master-worker version of the location algorithm, which checks them against the reference surfaces previously computed. This can be done because the Amazon S3 data storage is used by both services through the S3 service interface.

Output. The result of this system (as depicted in Fig. 1) is the identification of the activities the users carry out while in the mall, producing an event log such as the example shown in Fig. 4a. This information, consisting in the name of the activity being executed, the identification of the case —a visit of a customer, from the entrance to the end—, and the timestamp in which each activity took place, is later used by process mining techniques in order to perform the behavior analysis.

4 Process mining

With the explosion of process-related data, the behavioral analysis and study of real processes have become more popular. Process mining is a research field offering techniques to discover, monitor, and enhance real processes by extracting knowledge from event logs recording information of the executed activities [5]. Through this analysis, we can understand *what is really happening in a process* [4], and improve the process based on this information.

In this way, we can exploit the process mining potential to study and validate the customers' behavior by taking the information obtained with the localization system explained in Section 3. This section introduces some basic notions about the process mining field, and describes the details of the process mining analysis performed in this paper.

4.1 Process discovery

Process discovery plays a dominant role in all the stages of the process mining lifecycle [5]. The starting point is usually an event log recording the execution of the activities in the process, as formalized below.

Definition 1 (Event, Trace, and Event log). Given a set $\alpha \in A$ representing the activities of a process, an *event* ε represents a particular execution of the activity, storing information such as the start and end timestamps, the resource, cost, etc. A *trace* is a list (sequence) $\tau = \langle\ \varepsilon_1, ..., \varepsilon_n\ \rangle$ of events ε_i occurring at a time index i relative to the other events in τ. Each trace corresponds to an execution of the process, i.e., a process instance. We define an *event log* $L = [\tau_1, ..., \tau_m]$ as a multiset of traces τ_i.

Fig. 4a shows an example of an event log depicting 20 events of 7 different traces (trace 01 is depicted in Fig. 4d). With this information, process discovery techniques aim to build a process model —i.e. a social workflow— depicting the relations between the activities based on the behavior observed in the log.

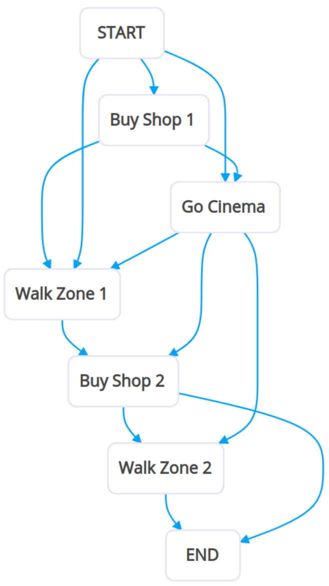
Definition 2 (Process model). Given a set $\alpha \in A$ representing the activities of a process, a *process model* is a tuple $M = (N, E)$ where $V \subseteq A \bigcup \{'Start','End'\}$ denotes the set of nodes in the process model representing the activities in the process, plus two artificial nodes to centralize the start and end of the process; and $E \subseteq N \times N$ denotes the set of edges representing a causal relation between two activities.

Fig. 4b shows an example of a process model depicting the behavior recorded in the event log in Fig. 4a. Discovering a good-quality and readable process model is crucial to maximizing the analysis and improvement of the process in the next stages. Nevertheless, in some scenarios where the executed activities of the process do not follow a clear structure, the discovered process model might be too complex to be analyzed directly.

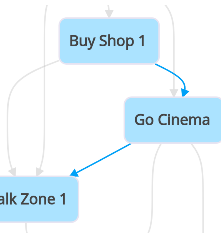
In these cases, post-processing techniques are used to extract as much information as possible from the process. A common practice is to focus on the parts of the model that are being executed more frequently, i.e., the frequent activity patterns.

Case	Activity	Timestamp
01	`Buy Shop 1`	2020-11-14 10:35:41
01	`Go Cinema`	2020-11-14 10:36:42
02	`Buy Shop 1`	2020-11-14 10:38:59
03	`Buy Shop 1`	2020-11-14 10:41:11
04	`Buy Shop 1`	2020-11-14 10:41:28
01	`Walk Zone 1`	2020-11-14 10:43:17
05	`Walk Zone 1`	2020-11-14 10:46:09
01	`Buy Shop 2`	2020-11-14 10:49:47
04	`Go Cinema`	2020-11-14 10:52:03
02	`Go Cinema`	2020-11-14 10:53:00
03	`Walk Zone 1`	2020-11-14 10:55:11
02	`Walk Zone 2`	2020-11-14 11:08:41
04	`Buy Shop 2`	2020-11-14 11:12:40
03	`Buy Shop 2`	2020-11-14 11:18:26
05	`Buy Shop 2`	2020-11-14 12:37:46
03	`Walk Zone 2`	2020-11-14 13:39:36
04	`Walk Zone 2`	2020-11-14 13:41:11
05	`Walk Zone 2`	2020-11-14 15:55:38
06	`Buy Shop 1`	2020-11-14 16:41:11
07	`Go Cinema`	2020-11-14 16:51:01
...	...	...

(a) Partial event log example.



(b) Process model corresponding log in Fig. 4a.



(c) Activity pattern example.

⟨ `Buy Shop 1`, `Go Cinema`, `Walk Zone 1`, `Buy Shop 2` ⟩

(d) Sequence of activities of the trace 01 of the log in Fig. 4a.

Fig. 4 Example of an event log, a trace, a process model and an activity pattern.

Definition 3 (Activity pattern). Given an event log L and a process model $M = (N, E)$, an *activity pattern* $P = (N', E')$ where $N' \subseteq N$, and $E' \subseteq E$, is a substructure of the process model depicting part of its behavior —a subprocess.

Fig. 4c depicts an example of an activity pattern of the event log and process model in Fig. 4. Given a frequency threshold, an activity pattern is said to be *frequent* when observed in a percentage of traces of the event log equal or higher than this threshold. For instance, for an event log of 100 traces, and a frequency threshold of 65%, the frequent activity patterns are those activity patterns observed in at least 65 traces. In this sense, the optimizations and improvements can be done w.r.t. the parts that are happening more often.

4.2 Conformance checking

Another main stage of the process mining lifecycle is conformance checking. Conformance checking takes action once both the event log – recording the observed behavior – and the process model – either discovered or designed by hand – are available. The aim of conformance checking is to relate the events in the event log with the activities in the process model, and compare them to detect commonalities and discrepancies between them [5]. This conformance analysis is useful to detect if reality, as recorded in the log, conforms to the model and vice versa [6]. For instance, by applying conformance checking techniques, a path designed in the process model not being observed in the event log can be detected. This detection allows the managers to inspect why this expected behavior is not being recorded, and act accordingly. Conversely, a conformance analysis can also detect a path observed in the event log but not modeled by the process model. This phenomenon can hint at a deviation in the real behavior that might be undesired – and hence, corrected –, or desired but unpredicted – and hence, added to the process model.

For instance, by applying conformance checking techniques to the process model depicted in Fig. 4b, we can confirm that it represents properly the behavior observed in the event log in Fig. 4a. The process model supports all the observed traces. Nevertheless, we can also detect that the process model allows for behavior not observed in the event log, e.g., the path ⟨ `Start`, `Go Cinema`, `Buy Shop 2`, `End` ⟩. As we can see, conformance checking improves the information about the process that can be extracted with the process model.

4.3 Process mining usage

As said before, for the context of this paper, we want to analyze the behavior of customers in the mall. Thus, we consider as the process' activities those activities that each user carries out while in the mall. In this sense, each full interaction of a customer in the mall, from the entrance to the end, represents a trace in the process. The event log consists of the collection of traces recording the itineraries customers have performed in the mall.

We have used conformance checking techniques to validate the recorded data by the smartphone application w.r.t. the ground truth. With the use of these techniques, a comparison of the ground truth process and the recorded data can be made, detecting existent deviations if any. These deviations have been further analyzed to detect the reason behind their existence.

We have used the tools provided by the InVerbis Analytics service [25] for process mining and visualization. These tools let us visualize graphs generated with the ProDiGen [45] algorithm to compute the process model depicting the customers' behavior recorded by the smartphone application and analyze it. ProDiGen is a genetic discovery algorithm[1] searching for complete, precise, and simple process models. Given an event log, ProDiGen first computes a simple heuristics-based process model and applies mutation operations to create the first population of the genetic procedure. Then, iteration by iteration, a new population is created by applying crossover and mutation operations to some selected process models from the previous population. This evolutionary procedure is repeated until a fixed number of populations is reached, or until a process model with enough quality is discovered. The result of the evolutionary process is the process model with the best quality measures among those created.

To further analyze the customers' behavior, and to extract activity patterns being executed frequently to focus on the most common behavior, we have used WoMine [11]. WoMine is an algorithm to extract frequent activity patterns – i.e., subprocesses that are executed frequently – from a process model, measuring their frequency in the instances of the log. One of the advantages of WoMine is that it can detect activity patterns with all types of structures: sequences, concurrency, choices, and loops. Furthermore, it can also ensure in which traces the frequent activity pattern is being executed. Given an event log and a process model, WoMine performs an *a priori* search starting with the activities of the process, and expanding them by adding relations and other activities to form subprocesses. On each iteration, the frequency of the computed subprocesses is calculated – using conformance techniques – to retain only the frequent ones, and optimize in this way the search. As a result, WoMine obtains the subprocesses of the process model with maximum size, ensuring that they have been executed by a number of users higher than a predefined threshold.

Output. One of the outputs of the multiple process mining analyses applied (as can be seen in Fig. 1) is the process model depicting the behavior of the customers. This process model is obtained by the discovery algorithm, and allows the analysts to inspect the overall behavior of the customers. Furthermore, this process model is also used by other process mining techniques to discover the frequent activity patterns —patterns of customers' frequent behavior—, which constitutes the other output. In addition, these analyses also produce insights about the process such as the verification of the recorded activities, which is obtained by applying conformance checking techniques, and the information given by the process model and the frequent activity patterns.

[1] A genetic algorithm is a metaheuristic based on the natural selection process. The process of a genetic algorithm starts with an initial population of individuals. Iteratively, some of the individuals in the current population are selected to breed the next one by applying mutation and crossover operations to them. After a fixed number of populations, the best quality individual is selected as the result.

5 Automatic Linguistic Reporting

In this section, we first introduce the field of Natural Language Generation (NLG). Then, we go in depth regarding how to automatically produce linguistic reports in the application context described in previous sections.

5.1 Revisiting NLG Approaches

NLG [40] provides models and methods for generating insights on data through natural language, producing textual descriptions that summarize the most relevant information of the data that is described. Natural language is an effective way of conveying information to humans because (1) it does not rely solely on the human capacity to identify or understand visual patterns or trends and (2) it may include uncertain terms or expressions, which are very effective for communication either alone or combined with other modalities [37].

Since the early developments in the 1980s, the complexity of NLG systems has notably increased and there are now several techniques and methodologies which guide the building of these solutions [20]. Also, different sub-fields have emerged within this area, such as generation of coherent texts from other texts, (interactive) dialog systems, computational creativity, or Data-to-Text systems [40], where texts are generated from numerical or symbolic data sets or series. Although the design of NLG systems is an open field, there exists some consensus about the basic tasks that NLG systems perform, being the traditional generic pipeline description in [41] the most commonly accepted. In general terms, the task of converting some input data into an output text can be divided into the following three sequential stages which convert step by step the (numerical) input data into natural language texts (see Fig. 5):

- Text planning, where the information to be conveyed in the text is identified (content determination), as well as some order and general structure of the text is planned.
- Sentence planning, which includes grouping of messages when needed (sentence aggregation), and decisions about the words/expressions to be used (referring expressions generation and/or lexicalization).
- Surface text realization, which consists of generating a syntactically, morphologically, and orthographically correct text.

Among these traditional approaches, template-based systems merge (part of) the previously described stages, and directly map the non-linguistic input to the final surface text realization. In [44] it is argued that, although template-based systems were initially deemed as inferior to standard NLG systems (mostly in terms of maintainability, or variation) they are able to perform all NLG tasks in a linguistically well-founded way. Furthermore, many successful template-based systems deviate from the prototypical fill-in-the blank template basic approaches, whilst there are examples of standard NLG systems

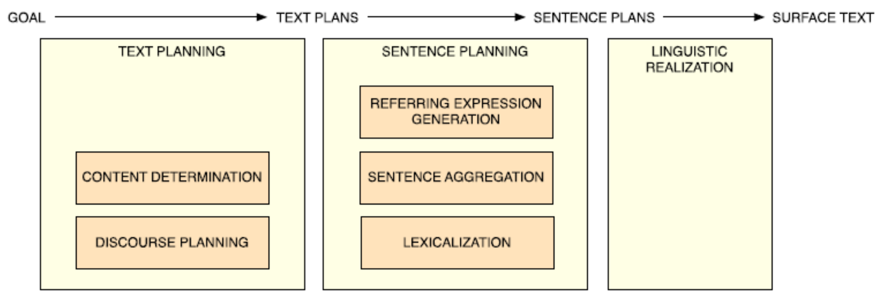


Fig. 5 Generic NLG system pipeline as depicted in [41].

which perform some tasks in a “less than well-founded fashion”. Nowadays, considering the different levels of complexity of NLG approaches and applications, this template-based vs standard NLG dichotomy is considered to be outdated and there is a preference to focus on the existing “different levels of sophistication in texts generation [16].”

Following a different approach, recent deep neural models for data-to-text generation are becoming popular as end-to-end approaches, which produce natural language texts from data without explicit intermediate representations in between [10, 20]. For some specific tasks, neural end-to-end models have proved to generate complex and rich outputs by learning from datasets [15], in spite of being affected by some undesired effects (e.g. hallucination, repetitions or lack of explainability and generalization when addressing unknown situations). It is also worth noting that traditional pipeline-based approaches usually generalize better to inputs which are not present in the datasets.

5.2 Linguistic Reports for Process Mining

As we have seen in Section 4, processes can be modeled and resultant models are usually represented in a graphical manner, by means of nodes and arrows which represent the activities that take place in a process as well as the dependencies among them. Sometimes, other properties of the process, such as temporal properties, the use and creation of data, decisions, and resources involved in the execution of the process, and so on, are also included in the models. These graphic representations are the common way to communicate relevant information about processes to users. Also visual analytics (or data visualization) [3] techniques are popular, focusing on data analysis. These approaches ultimately rely on human capabilities to understand and identify valuable information from the data. However, process models are often very complex, and both process models and visual analytics may be very difficult to be understood by non-trained users. Within this context, natural language descriptions can be a good approach to enable or enhance the understanding of process models and visual analytics, as they can summarize, combine and communicate information in ways it would not be possible only with visual representations. Since natural language is the inherent way of communicating for humans, this modality of information conveying is likely to be more easily understood and consumed by users. In this way, conveying useful process models information would not only rely on the user capabilities to identify or understand patterns, trends, and so on, from visual representations.

Moreover, research in other areas suggests that knowledge and expertise are required to understand graphical information [35] and domain experts can make better decisions when graphical displays are combined with textual descriptions [26, 34]. In this regard, transforming process models into textual descriptions has been identified [2] as one of the recent challenges for the application of Natural Language Processing in the field of process mining and analytics.

Although NLG has shown its scientific and technological maturity in many fields of application, only recently NLG has been applied to process mining, for generating insights on process data through natural language. The approaches in the related literature follow these two perspectives:

– Control-Flow perspective models align the graph-based representation of a process model with its textual description [1,29,34,42,43]. In these models, relationships between the activities of the process model have a textual description, which usually includes references to the resources that perform these activities.
– Case Description techniques focus on generating textual descriptions about the execution of single activities or activity sequences [14], therefore considering neither the process model structure nor the relations between activities.

Another recent proposal [19] presented a framework where three paradigms were integrated: (1) process analytics for extracting temporal and structural information from a process; (2) fuzzy linguistic protoforms for modeling uncertain terms [38]; and (3) NLG for building the explanations. We will not consider here other approaches which heavily rely on classical data mining techniques and only use event log data as their inputs, paying no attention to the underlying process model.

Our approach for Automatic Linguistic Reporting, associated to process model discovery, is made up of the following stages:

– Content determination: after having discovered the process model, several numerical indicators are evaluated in order to extract relevant information which will be conveyed in the textual descriptions. Most of them are temporal data regarding traces, such as the average duration of the complete process, the most relevant deviations, as well as the duration of some key activities (e.g., we pay special attention to the longest activities).
– Discourse planning: we have implemented a discourse planning structure which has proven successful in many other real applications of NLG [39]. Firstly, a general description is presented, providing quantitative information about the average duration of the process and the number of cases

which conform temporal-related information about the duration of the activities. This information is relevant in order to provide a general overview of the complete process duration. Following the general description, the number of cases that deviate from the average duration are highlighted. Two types of nuances are quantified here: the number of cases which have a longer duration than the average and the number of cases which have a shorter duration. The duration of the cases which correspond to both nuances are linguistically compared to the general case using relative comparison expressions (e.g., "clearly longer", "much shorter", etc.) in order to provide the reader with an intuition about to what extent cases deviate but, at the same time, avoiding numerical details. Finally, relevant information for the user indicating what aspects he/she should pay attention to is provided: (1) a list of the specific activities with longer duration or higher temporal deviation from the average; and (2) a list of the transitions between activities to be revised.
– Linguistic realization: since our approach is a template-based one, all the transformations described in the Sentence planning stage in Figure 5 are merged in the surface text realization.

Among the several technological choices for performing the linguistic realization stage of the textual reports, we used the InVerbis Analytics service [25], as will be described in detail in Section 6.4. InVerbis is an integral cloud-based process analytics platform, which provides several process mining features such as the discovery of process models and variants, loop detection, and conformance analysis against predefined process models, as well as a discovery report that includes standard process analytics, such as visualizations of duration and frequency of process variants, activities, and transitions. Discovery reports include automatically generated textual narratives that complement the visualizations and suggest potential bottlenecks within the process from a global perspective and directly produce the previously indicated text planning structure. A simplified example showing these elements in a real case is the following one:

The process has an average duration of 10.3 minute(s) with a deviation of ±10.3 minutes. 50 of the analyzed cases conform to this average. The remaining 9 are composed of 3 cases that can be considered clearly shorter, and 6 clearly longer [...]. It is recommended that at least the following activities are checked, since their duration and/or temporal duration is excessive: Eating in the Restaurants area at a coffee shop, ... Likewise, the following transitions between activities should be revised: Walking outdoors - Window shopping outdoors at restaurant, ...

The complete example including the full natural language report of a real process, as well as the visual information, will be later presented, discussed, and evaluated in Section 6.4 and Figure 10.

Output. Following with the pipeline depicted in Fig. 1, the application of the NLG techniques commented in this section produces human-friendly linguistic descriptions of insights about the customers' behavior analyzed with process mining techniques.

6 Test-bed and results

In this section, we present the use case experimentation performed in this work, which consists of the customer behavioral analysis in a mall, and the automatic linguistic reporting of this analysis.

6.1 Test-bed description

We have developed our system in the open mall *"C.C. La Dehesa"*, located in Alcalá de Henares (Madrid). This open mall (see Fig. 6) has 36.000m^2 of outdoor shopping (green area; which includes fast food restaurants and shops) and indoor shopping composed by three big zones: 65 *shops* (orange area), one big *hypermarket* (blue area) and 7 *restaurants* (yellow area).

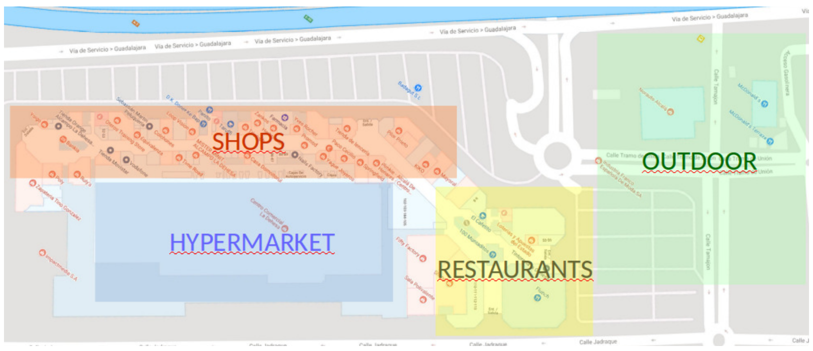


Fig. 6 Open mall "C.C. La Dehesa".

It is worth noting that in the indoor shops' zone, the ceiling is made by glass, permitting to obtain localization by GPS while in the rest only WiFi localization is available. In the outdoor area, GPS and WiFi are available. This test-bed was selected because it fits perfectly with the definition of open-mall, shops core in indoor-outdoor with mixed localization available.

We used our previously developed application to train and test the complete system running in three different mobile devices: two smartphones (Samsung Galaxy S6 and S6 edge) and one tablet (Samsung Galaxy Tab 2). The application was used in two ways: (1) the *training dataset* was collected by means of saving several *paths* at static places, a path consists of frames at key points, and frames are formed by sensors' data and activities at zones and specific places introduced by an expert user; and (2) the *test dataset* was collected by means of saving several paths with the user in movement. The activity, zone and place information introduced by the user for the *test dataset* will be used as ground truth in the validation stage.

Table 1 shows the 6 paths that we have considered in order to validate our system. Paths consist of 4 or 5 steps of activities that start in a tuple activity-zone-place or key point and evolve to the next one.

Table 1 Paths in Shopping Mall.

			App waypoint			
Path	**Device**	**Distance**	**Step**	**Activity**	**Zone**	**Place**
00	**S6 /S6E**	**300 m**	1	Eating	Restaurants	CoffeeShop
			2	Walking	Restaurants	No Place
			3	Walking	Shops	No Place
			4	Shop window	Shops	ToyShop
01	**S6**	**410 m**	1	Eating	Restaurants	CoffeeShop
			2	Walking	Restaurants	No Place
			3	Walking	Shops	No Place
			4	Walking	Hypermarket	No Place
			5	Buying	Hypermarket	No Place
02	**Tab2**	**250 m**	1	Eating	Restaurants	CoffeeShop
			2	Walking	Restaurants	No Place
			3	Walking	Outdoor	No Place
			4	Shop window	Outdoor	Restaurant
			5	Buying	Outdoor	Restaurant
03	**Tab 2 / S6**	**300 m**	1	Shop window	Restaurants	CoffeeShop
			2	Walking	Restaurants	No Place
			3	Walking	Shops	No Place
			4	Shop window	Shops	ToyShop
04	**Tab 2 / S6E**	**410 m**	1	Shop window	Restaurants	CoffeeShop
			2	Walking	Restaurants	No Place
			3	Walking	Shops	No Place
			4	Walking	Hypermarket	No Place
			5	Buying	Hypermarket	No Place
05	**S6 / S6 E**	**250 m**	1	Shop window	Restaurants	CoffeeShop
			2	Walking	Restaurants	No Place
			3	Walking	Outdoor	No Place
			4	Shop window	Outdoor	Restaurant
			5	Buying	Outdoor	Restaurant

S6: Samsung Galaxy S6
S6 E: Samsung Galaxy S6 Edge
Tab 2: Samsung Galaxy Tab 2

The different paths were executed several times and on several dates to validate the results. As a result, we collected data from 17 paths for training (17 static places) and 35 paths for testing (almost 6 times per path in movement). The table also shows the approximated traveled distance at each test's path, which ranges from 250m to 410m.

As an illustrative example, **Path 00** corresponds to about 300 m of travelled distance in about 31 minutes. It includes the next activities:

1. *Eating* (having the breakfast) at *Place* CoffeeShop in *Restaurants* zone during 20 minutes.
2. *Walking* along *Restaurants* zone during 2 minutes.
3. *Walking* along *Shops* zone during 5 minutes.
4. Looking at *Shop window* of ToyShop *Place* at *Shops* zone during 4 minutes.

As you can see, the difference between activities 2 and 3 was the zone, key information that we obtained from the localization system.

The summary of high-level activities that we have added to the basic activities recognized by Google's API are:

– Eating: we suppose that person is eating when he/she is localized at a restaurants' zone during a higher time than a threshold.
– Walking: accelerometers and localization are key features to determine that person is walking.
– Shop window: this activity means that a person is standing at the front of a shop but lower time than a threshold.
– Buying: localization is a key feature to determine that person is buying inside a shop or hypermarket.

6.2 Analysis of the experimentation conducted with the cloud-based implementation

Due to the fact that the services shown in Fig. 3 are deployed in a cloud infrastructure, it is necessary to select the computing resources to be provisioned according to the BAI4SOW applications' requirements. Let us now briefly analyze the execution times and costs of this approach.

In order to conduct this evaluation, we have used the data reported in [24] of a user who has freely walked in the environment described previously (a medium-size indoor scenario of around 2.500 m^2). During the walk, the user's mobile has measured 300 RSS values in different positions. Also, a location request has been sent for each of these values to determine her/his location. Previously, 100 reference surfaces with an accuracy of 1 cm were computed to be used by the location method in order to increase significantly the number of cells to be processed for the same surface (25 million cells in the scenario of 2,500m^2). Besides, all the detected APs (a total of 105) and, as consequence, the 105 surfaces have been used to perform the localization. Two different trajectories were used to test the localization system, obtaining mean localization errors around 1,5 meters.

As it was previously detailed, it took 44.4 hours to execute the sequential version of the algorithms to compute all the surfaces during the training stage, and 325 seconds to determine the current location of a customer using a server with 8 Intel i7-4790K CPUs at 4.00GHz, a local SSD disk and 32GB of RAM. Note that the experiment had no additional costs apart from the requirement of owning the required hardware and a correct configuration/administration of the operating system and the software. Now, different deployments of the architecture depicted in Section 3.2 are going to be configured and analyzed using the *Amazon Elastic Compute Cloud* infrastructure (Amazon EC2), which offers a comprehensive and complete elastic Web-Scale computing catalog of computing resources for different purposes.

The experimentation is composed of two phases. First, the time and costs to create the surfaces are measured in $Phase_1$. After that, the time and cost to serve the location requests and complete the experiment for each of the proposed deployments is measured in $Phase_2$. The time per request can be calculated by dividing *Time* $Phase_2$ by 300.

Table 2 shows the execution times and costs of the different configuration deployments using the *Amazon EC2* infrastructure. *R5 instances* have been chosen as the best option because they are optimized for memory-intensive applications and they can offer good performance for this type of processing problems. Also, R5 instances offer a better relation between memory per CPU and cost with respect to R4 instances. The on-demand hiring mechanism has been used for the experiments, and the number of cores, memory, and costs of the instances used are specified in Table 2 as well for the North Virginia (USA) region.

Table 2 Evaluation of execution time and costs for a cloud-based deployment of the algorithms.

EC2 instance	**vCPU**	**Memory (GiB)**	**Cost/hour (USD)**	**Time $Phase_1$ (min)**	**Time $Phase_2$ (min)**	**Total time (h)**	**Total cost (USD)**
Sequential	8	32	-	1085.2	1579.1	44.4	-
Concurrent deployment							
r5.xlarge	4	32	0.252	161.3	783.6	15.7	4.03
r5.2xlarge	8	64	0.504	126.4	398.9	8.8	4.54
r5.4xlarge	16	128	1.008	102.3	202.2	5.1	6.05
r5.8xlarge	32	256	2.016	83.4	103.1	3.1	8.06
r5.16xlarge	64	512	4.032	66.72	53.4	2.0	12.01
Distributed deployment							
3x r5.16xlarge	192	1536	12.096	24.9	22.1	0.8	12.01
6x r5.16xlarge	384	3072	24.192	13.2	11.5	0.4	24.19

First, four R5 instances were used to conduct the concurrent execution of the algorithms. We started with a r5.xlarge instance (4 vCPUs and 32 GiB of memory) and twice the configuration up to a r5.16xlarge instance (64 vCPUs and 512 GiB of memory). As it can be seen in Table 2, the execution time in these instances presents a linear speedup according to the configuration of the machines (vCPUs and memory). The best execution time is then obtained by the r5.16xlarge instance, which took 120.11 minutes to complete the experimentation. This means that the execution time is improved more than an order of magnitude with respect to the time of the sequential version (44.4 hours).

The cost of experiments (column *Total costs*) considers the costs of computing resources and of storing the data into the *Amazon S3 service*. This is calculated by multiplying the cost per hour of the selected instances (column *Cost/hour*) by the total hours of execution time (column *Total time*). The cost of storing the data is the same for all the configurations, as the size of the used data is around 66 GB and the storage price is 0.023 USD/GB for the first 50TB/month. Therefore, the cost of the execution is proportional to the configuration of the instances in terms of the number of cores and memory.

Finally, we also conducted the experimentation using two different distributed environments. These environments are composed of a pool of three and six *r5.16xlarge* instances, respectively. The worker nodes have been distributed between the vCPUs of the different instances, getting a maximum of 191/382 workers running simultaneously on the provisioned instances. The master is deployed on the remaining core.

Table 2 also depicts the results obtained for each configuration in the distributed environments. As shown, the execution time of the experiment continues following a linear speedup according to the configuration of the environment. The overhead due to the parallelization is minimum and the experiment can be completed in less than 25 minutes with a pool of six instances. However, as the cost of each instance is per hour, the cost of using six instances is twice the cost of using three (24.19 USD vs 12.01 USD, respectively), as with both configurations the experiment can be carried out in less than an hour. Therefore, the relationship between the need to complete the execution in less than 25 minutes versus the additional cost it requires must be carefully evaluated.

Independent of the results shown in the table, the main conclusion of this experiment is that the master-worker architecture can be easily deployed in a cloud environment by applying different configurations (we are selected some of these configurations on the basis of our experience, but other different could be also considered) and that it is a good option to accelerate the location algorithms proposed in the paper. The architecture allows the process of location to be decomposed as much as possible and takes the advantage of the scalable nature of the cloud in order to optimise that process.

6.3 User Behavioral Analysis

This section describes how we have applied process mining techniques to the activity and localization data obtained and described in the previous sections. The purpose of this analysis is twofold. First, to validate the recorded behavior using conformance checking techniques by comparing it against the ground truth data. Second, to inspect the customers' behavior by applying process mining algorithms to the recorded event log, and to extract insights useful to the mall managers.

The first step has been to build the process model (depicted in Fig. 7) describing the six paths of the ground truth in Table 1. With this model and the recorded event log, we have executed conformance checking techniques to analyze the deviations between the captured traces and the expected behavior.

We have found that 79.66% of the recorded traces have shown no deviations w.r.t. the ground truth model, proving that they have been recorded correctly by the localization system and smartphone application. On the other hand, the remaining 20.34% presented some deviations in the analysis. Two different cases have been detected regarding these traces. On the one hand, the deviations present in 11.86% of the traces have been detected as missing events from the ground truth paths, i.e., the recorded data corresponds to partial paths of the ground truth. In these cases, the most probable reason is a stop in the WiFi signal, either in the mall receptors or in the users' devices,

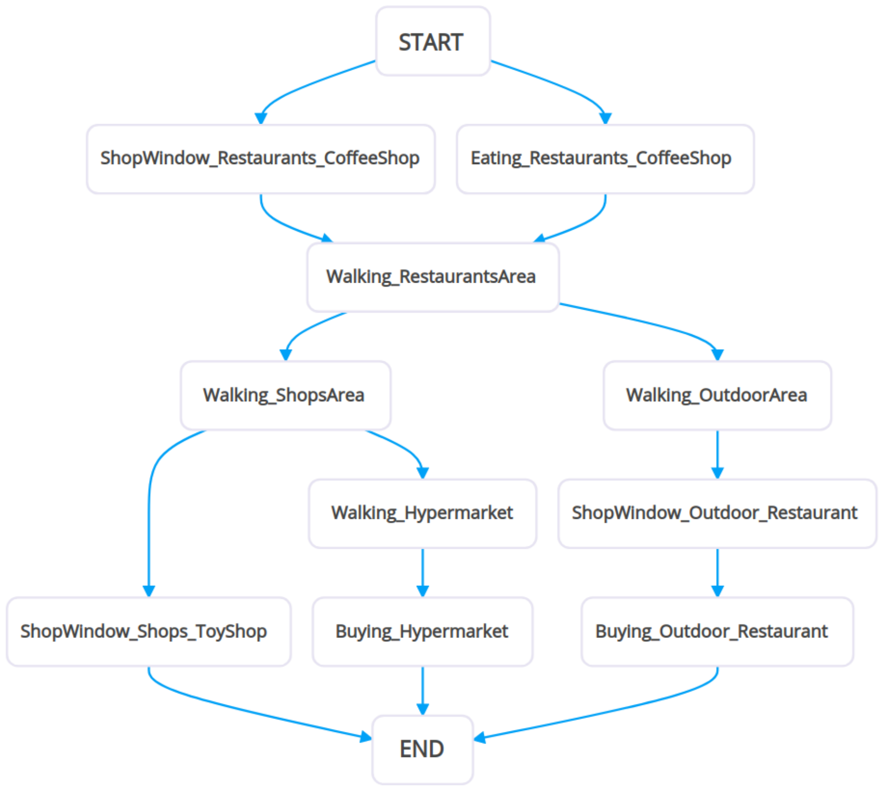


Fig. 7 Process model depicting the behavior of the ground truth.

causing them to not record all events in the path. The remaining 8.48% of the traces have shown a deviation regarding the localization system. In these cases, a full path in the ground truth was recorded, but some of the events had been mislocalized. For instance, one of the traces corresponded with path 05, but recording the event 'Buying Outdoor Burger' as 'Buying Restaurants Burger', probably due to the proximity of the locations.

After analyzing the recorded behavior w.r.t. the ground truth, we have discovered the process model depicting the users' recorded behavior to extract insights about their behavior, and to show the potential of applying process mining techniques to this scenario. Fig. 8 shows the process model obtained with ProDiGen [45]. We observe that the behavior of users doesn't have a clear structure. Few conclusions can be extracted from an unstructured process as the one shown in Fig. 8. Nevertheless, this result is not surprising, as the users' behavior in a broad area such as a mall is expected to be unstructured.

To cope with this shortcoming, a more specific process mining analysis can be run, focusing on the parts of the process that the users are executing more frequently. We have used WoMine [11] to extract the frequent activity patterns – subprocesses – modeling the behavior commonly executed by

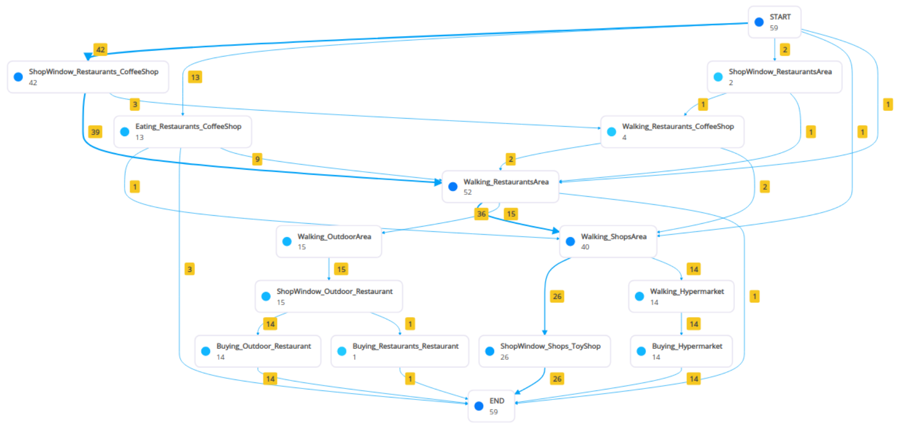
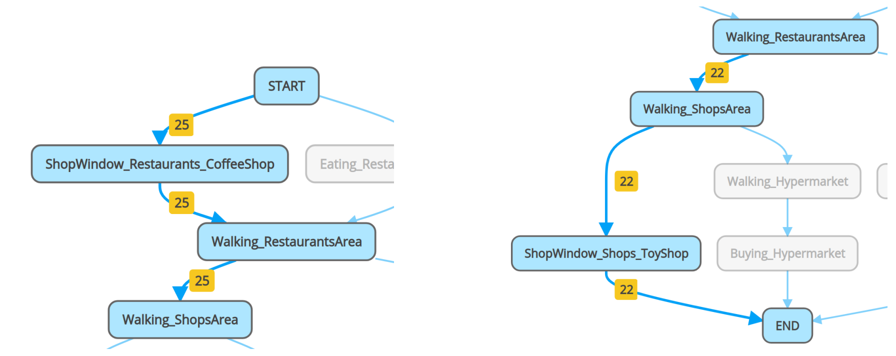

START
59
ShopWindow_Restaurants_CoffeeShop
42
ShopWindow_RestaurantsArea
2
Eating_Restaurants_CoffeeShop
13
Walking_Restaurants_CoffeeShop
4
Walking_RestaurantsArea
52
Walking_OutdoorArea
15
Walking_ShopsArea
40
ShopWindow_Outdoor_Restaurant
15
Walking_Hypermarket
14
Buying_Outdoor_Restaurant
14
Buying_Restaurants_Restaurant
1
ShopWindow_Shops_ToyShop
26
Buying_Hypermarket
14
END
59

Fig. 8 Process model discovered by ProDiGen [45] depicting the behavior of the customers in the mall.

the users. Fig. 9 shows two of the more relevant frequent activity patterns extracted by WoMine. Fig. 9a highlights a behavioral pattern with a frequency of 42%. This behavior has been conducted by the 42% of the recorded customers, who have started in the shopwindow of the coffee shop, walking through the restaurants' area, and later through the shops' area without buying anything. This pattern can hint at a need in the Coffee Shop to make changes in order to engage more customers who are doing window shopping (i.e, they stop briefly to look at the shop window but leave to other areas without coming in the shop). Fig. 9b depicts another behavioral pattern being executed by 37% of the customers. This pattern shows that 1 out of 3 customers pass walking through the restaurants and shops areas, leaving the mall after staring at the



(a) Frequent pattern modeling the behavior performed by a 42% of the customers.

(b) Frequent pattern modeling the behavior performed by a 37% of the customers.

Fig. 9 Frequent patterns (highlighted structures) extracted by WoMine using the event log recording the activities of the customers in the mall.

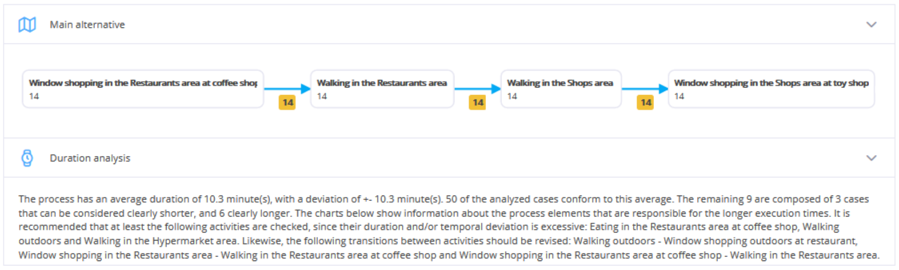
shop window of the toy shop. This pattern hints that it might also be interesting to work on the ads at the more frequented areas, Restaurants, and Shops, to deal with those customers who leave the mall without consuming anything.

In summary, the process mining analysis allows to discover the process model depicting the behavior of the customers, and to inspect in more detail their behavior through the frequent activity patterns. The behavior modeled by the two reported patterns – which were conducted by almost half of the customers – was useful to detect possible redesigns in the marketing campaign of some stores to increase the attention of the customers. Furthermore, to improve the knowledge and insights obtained from the analysis, automatic linguistic reports can be extracted from this discovered behavior, as we will describe in the next section.

6.4 Automatic linguistic reporting evaluation and results

We applied NLG techniques to automatically generate human-friendly linguistic reports associated with the previous process mining analysis. It is worth noting that both process mining (see Section 4) and report generation (see Section 5) were connected with the help of tools provided by the InVerbis Analytics service [25].

More precisely, we considered a sequential NLG pipeline for linguistic reporting, as depicted in Fig. 5 in Section 5. We show in Fig. 10 the report generated for the test bed under consideration.

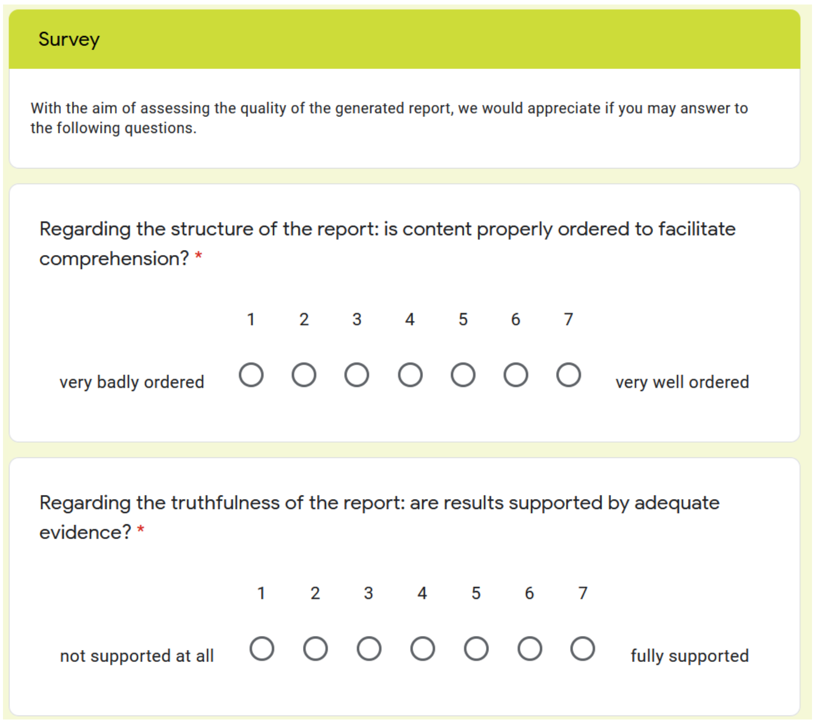

Main alternative
Window shopping in the Restaurants area at coffee shop
14
14
Walking in the Restaurants area
14
14
Walking in the Shops area
14
14
Window shopping in the Shops area at toy shop
14
Duration analysis

The process has an average duration of 10.3 minute(s), with a deviation of +- 10.3 minute(s). 50 of the analyzed cases conform to this average. The remaining 9 are composed of 3 cases that can be considered clearly shorter, and 6 clearly longer. The charts below show information about the process elements that are responsible for the longer execution times. It is recommended that at least the following activities are checked, since their duration and/or temporal deviation is excessive: Eating in the Restaurants area at coffee shop, Walking outdoors and Walking in the Hypermarket area. Likewise, the following transitions between activities should be revised: Walking outdoors - Window shopping outdoors at restaurant, Window shopping in the Restaurants area - Walking in the Restaurants area at coffee shop and Window shopping in the Restaurants area at coffee shop - Walking in the Restaurants area.

Fig. 10 Linguistic report in natural language generated using [25] for the real case considered.

As indicated in Section 5, this report pays special attention to the most frequent pattern (see Fig. 9a) extracted from the graph depicted in Fig. 8. The duration analysis in the report also refers to activities (and transition between activities) that deserve special consideration.

For the first time in the application of NLG to process mining and analytics (at least to the best of our knowledge), we have conducted a human expert validation of the natural language reports in a real scenario. We carried out an online survey that follows the best practices in NLG evaluation, with the aim of evaluating the quality of the generated linguistic report (see a sketch of the questionnaire in Fig. 11).


Survey

With the aim of assessing the quality of the generated report, we would appreciate if you may answer to the following questions.

Regarding the structure of the report: is content properly ordered to facilitate comprehension? *

	1	2	3	4	5	6	7	
very badly ordered	○	○	○	○	○	○	○	very well ordered

Regarding the truthfulness of the report: are results supported by adequate evidence? *

	1	2	3	4	5	6	7	
not supported at all	○	○	○	○	○	○	○	fully supported

Fig. 11 Sketch of the questionnaire for assessing quality of linguistic reporting.

The questionnaire is inspired by the so-called Gricean conversational maxims which were first presented in [23]: (G1) Maxim of Quantity ["Make your contribution as informative as required for the current purposes of the exchange. Do not make your contribution more informative than is required"]; (G2) Maxim of Quality ["Do not say what you believe to be false. Do not say that for which you lack adequate evidence"]; (G3) Maxim of Relation "Be relevant"]; (G4) Maxim of Manner ["Avoid the obscurity of expressions, avoid ambiguity, be brief (avoid unnecessary prolixity), be orderly"].

These conversational maxims were studied by Dale and Reiter in the context of NLG systems [13]. Moreover, they were also considered in previous publications (e.g., [12]) for evaluating the quality of NLG systems. It is worth noting that, in the context of NLG, human evaluation is considered as the most suitable evaluation method [27], since it is able to capture adequacy, correctness, and fluidity of the texts, among other relevant features.

Following the usual methodology in this field, we set up a questionnaire which is made up of six questions with the aim of covering properly all Gricean maxims. With the aim of minimizing the bias effect due to question order, the six questions were presented with shuffled order to each assessor. In addition, the answer to each question is given on a 7-point Likert scale [30]. This is a discrete scale where the assessor is expected to select the level of agreement with the given statement or question (e.g., from strongly disagree to strongly agree).

The target audience was made up of people who may be somehow familiar with the problem under consideration, i.e., intelligent data analysis and linguistic description of estimated customer activity patterns in open malls. Accordingly, we sent the questionnaire to attendants of the 21st International Workshop of Physical Agents (WAF) where our preliminary work [33] was presented.

We collected 41 answers to each question. Table 3 summarizes the computed scores. The first column identifies the correspondence between the questions given in the second column and the Gricean maxims. There is one question related to each of the three first maxims and three additional questions to cover the fourth maxim. We included three questions related to the last Gricean maxim for the sake of completeness, and looking for evaluating exhaustively the use of the "everyday natural language" in the NLG templates used for linguistic realization. Thus, we asked about the order of the given pieces of information, about the correct use of vocabulary, and about the general linguistic quality of the report. Results are encouraging, being 6 the median of answers for all questions. In addition, mean values are always above 5 and standard deviation (Std.) is below 1.5 in all cases. Even more relevant is the fact that the assessors globally indicate a very high degree of satisfaction with the generated report, since the general linguistic quality of the report (see Fig. 12) achieves a value of 6.17 in mean (with 0.76 std.), as remarked in bold in the last row in Table 3. This is the best indicator of the high quality of the texts generated automatically from the process data.

Table 3 Evaluation of linguistic reporting quality (7-point Likert scale).

Gricean Maxim	Question	Median	Mean	Std.
Quantity	Regarding the length of the report: is it adequate with respect to content?	6	5.51	1.15
Quality	Regarding the truthfulness of the report: are results supported by adequate evidence?	6	5.47	1.45
Relation	Regarding the relevance of the report: is the content relevant to the problem (estimating customer activity patterns in the mall)?	6	5.66	1.42
Manner	Regarding the structure of the report: is content properly ordered to facilitate comprehension?	6	5.49	1.13
	Regarding the clarity of the report: is the language correctly used?	6	6.10	0.91
	Regarding the general linguistic quality of the report: how good are grammar, syntax, orthography, and punctuation?	**6**	**6.17**	**0.76**

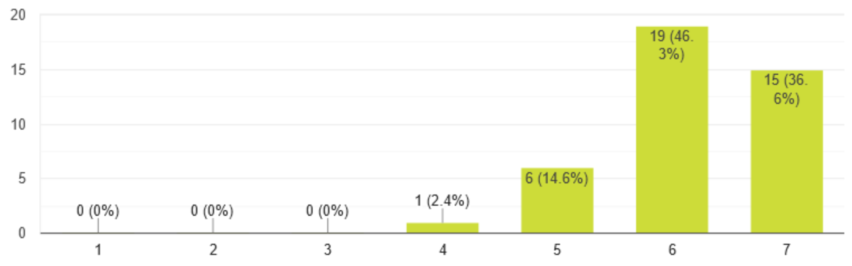


Fig. 12 Distribution of answers to the question about the general linguistic quality of the report.

7 Conclusions and Future work

We have presented a method to estimate the activity patterns associated with customers in open malls. Such patterns are based on localization data which are provided by a specific smartphone application for logging information from sensors and process mining techniques to identify what kind of usual activities are made by customers. We use the localization key information joined with data mining, parallelization, and monitoring techniques.

Our system has been tested in a challenging real scenario such as an open mall (mixed classical mall and isolated shops in the street) where location is obtained by mean of GPS and a WiFi localization system that is helped by a parallel computing method to speed up the whole process. From these localization data, we apply state of the art process mining techniques to first generate a process model (social workflow) describing the behavior made by customers in the mall. We have also analyzed the most common activity patterns performed by customers. The hints provided by these process mining techniques may help managers to take specific actions considering the customers' behavior and the activities they perform while in the mall.

We have also integrated an automatic linguistic reporting module, which automatically describes in natural language the most relevant characteristics of the process, providing complementary information to the one conveyed by the visualization of the process model. Following the best practices for human evaluation of natural language generation systems, 41 experts validated this approach, with a global score of 6.17 on average (0.76 of standard deviation) in a 7-point Likert scale. As far as we know, this is the first time that this type of evaluation is conducted in the field of NLG applied to process mining and analytics reporting in a real case. These results empirically validated the adequacy of automatic natural language textual reporting as a useful tool to facilitate understanding of the usual graphical outputs given by process mining visualization tools. We strongly believe that automatic linguistic reporting and process mining visualization tools can act as complementary tools to provide managers with valuable and meaningful insights no matter their technical or mathematical background.

This work has set a base in the analysis of customer behavior in open malls. Nevertheless, the analyses performed in this publication only take into account the control-flow information of the process, i.e. the sequence of activities that each user carries out. More interesting insights can be extracted if the analyses are performed taking into account other data related to the users such as age, gender, language, shopping interests, etc. Regarding process mining, many techniques have been proposed to perform analysis taking into account this type of personal data, known as multi-perspective process mining [28, 31]. However, personal data must be carefully considered to avoiding bias as well as respecting privacy and ethical issues [17, 18]. As future work, we plan to extend the mobile app to record properly this kind of information, and perform more precise analyses taking into account more attributes about the customers, while complying with legal and ethical issues.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science Innovation and Universities and the European Regional Development Fund (ERDF/FEDER) Grants RTI2018-099646-B-I00, TIN2017-84796-C2-1-R, TIN2017-90773-REDT, RED2018-102641-T and RYC-2016-19802 (Ramón y Cajal program, José M. Alonso). Also by the Galician Ministry of Education, University and Professional Training and the ERDF/FEDER program (ED431F2018/02, ED431C2018/29, ED431G2019/04 grants).

References

1. van der Aa, H., , Leopold, H., Reijers, H.A.: Detecting inconsistencies between process models and textual descriptions. In: Proceedings BPM 2015, *LNCS*, vol. 9253, pp. 90–105. Springer (2015)
2. van der Aa, H., Carmona, J., Leopold, H., Mendling, J., Padró, L.: Challenges and opportunities of applying natural language processing in business process management. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 2791–2801. Association for Computational Linguistics, Santa Fe, New Mexico, USA (2018). URL https://www.aclweb.org/anthology/C18-1236
3. van der Aalst, W.: Process Mining: Discovery, Conformance and Enhancement of Business Processes, vol. 136. Springer-Verlag Berlin Heidelberg (2011). DOI https://doi.org/10.1007/978-3-642-19345-3
4. van der Aalst, W.M.P.: Process Mining - Discovery, Conformance and Enhancement of Business Processes. Springer (2011). DOI 10.1007/978-3-642-19345-3. URL https://doi.org/10.1007/978-3-642-19345-3
5. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016). DOI 10.1007/978-3-662-49851-4. URL https://doi.org/10.1007/978-3-662-49851-4
6. van der Aalst, W.M.P., Adriansyah, A., de Medeiros, A.K.A., Arcieri, F., Baier, T., Blickle, T., Bose, R.P.J.C., van den Brand, P., Brandtjen, R., Buijs, J.C.A.M., Burattin, A., Carmona, J., Castellanos, M., Claes, J., Cook, J., Costantini, N., Curbera, F., Damiani, E., de Leoni, M., Delias, P., van Dongen, B.F., Dumas, M., Dustdar, S., Fahland, D., Ferreira, D.R., Gaaloul, W., van Geffen, F., Goel, S., Günther, C.W., Guzzo, A., Harmon, P., ter Hofstede, A.H.M., Hoogland, J., Ingvaldsen, J.E., Kato, K., Kuhn, R.,

Kumar, A., Rosa, M.L., Maggi, F.M., Malerba, D., Mans, R.S., Manuel, A., McCreesh, M., Mello, P., Mendling, J., Montali, M., Nezhad, H.R.M., zur Muehlen, M., Munoz-Gama, J., Pontieri, L., Ribeiro, J., Rozinat, A., Pérez, H.S., Pérez, R.S., Sepúlveda, M., Sinur, J., Soffer, P., Song, M., Sperduti, A., Stilo, G., Stoel, C., Swenson, K.D., Talamo, M., Tan, W., Turner, C., Vanthienen, J., Varvaressos, G., Verbeek, E., Verdonk, M., Vigo, R., Wang, J., Weber, B., Weidlich, M., Weijters, T., Wen, L., Westergaard, M., Wynn, M.T.: Process mining manifesto. In: F. Daniel, K. Barkaoui, S. Dustdar (eds.) Business Process Management Workshops - BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I, *Lecture Notes in Business Information Processing*, vol. 99, pp. 169–194. Springer (2011). DOI 10.1007/978-3-642-28108-2_19. URL https://doi.org/10.1007/978-3-642-28108-2_19

7. Amazon S3 - Simple Cloud Storage Service: https://aws.amazon.com/s3/. [Online; accessed in January 2021]
8. Amazon Web Services EC2 - Simple Cloud Hosting: https://aws.amazon.com/ec2/. [Online; accessed in January 2021]
9. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Pattern-Oriented Software Architecture, Volume 1, A System of Patterns. Wiley (1996)
10. Castro Ferreira, T., van der Lee, C., van Miltenburg, E., Krahmer, E.: Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 552–562. Association for Computational Linguistics, Hong Kong, China (2019). DOI 10.18653/v1/D19-1052. URL https://www.aclweb.org/anthology/D19-1052
11. Chapela-Campa, D., Mucientes, M., Lama, M.: Mining frequent patterns in process models. Inf. Sci. **472**, 235–257 (2019). DOI 10.1016/j.ins.2018.09.011. URL https://doi.org/10.1016/j.ins.2018.09.011
12. Conde-Clemente, P., Alonso, J.M., Nunes, E., Sanchez, A., Trivino, G.: New types of computational perceptions: Linguistic descriptions in deforestation analysis. Expert Systems with Applications **85**, 46–60 (2017). DOI 10.1016/j.eswa.2017.05.031
13. Dale, R., Reiter, E.: Computational interpretations of the gricean maxims in the generation of referring expressions. Cognitive Science **19**, 233–263 (1995)
14. Dijkman, R.M., Wilbik, A.: Linguistic summarization of event logs: A practical approach. Information Systems **67**, 114–125 (2017)
15. Duek, O., Novikova, J., Rieser, V.: Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. Computer Speech & Language **59**, 123–156 (2020). DOI https://doi.org/10.1016/j.csl.2019.06.009. URL https://www.sciencedirect.com/science/article/pii/S0885230819300919
16. Ehud Reiter: Nlg vs templates: Levels of sophistication in generating text. https://ehudreiter.com/2016/12/18/nlg-vs-templates/ (2016). [Online; accessed in April 2021]
17. EU High Level Expert Group on AI: AI Ethics Guidelines for Trustworthy AI. Tech. rep., European Commission, Brussels, Belgium (2019). DOI 10.2759/346720
18. EU High Level Expert Group on AI: The assessment list for trustworthy artificial intelligence (altai) for self assessment. Tech. rep., European Commission, Brussels, Belgium (2019). DOI 10.2759/002360
19. Fontenla-Seco, Y., Lama, M., Bugarín, A.: Process-to-text: A framework for the quantitative description of processes in natural language. In: F. Heintz, M. Milano, B. O'Sullivan (eds.) Trustworthy AI - Integrating Learning, Optimization and Reasoning, pp. 212–219. Springer International Publishing, Cham (2021)
20. Gatt, A., Krahmer, E.: Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. J. Artif. Intell. Res. **61**, 65–170 (2018). DOI 10.1613/jair.5477. URL https://doi.org/10.1613/jair.5477
21. GNU Octave: https://www.gnu.org/software/octave/index. [Online; accessed in January 2021]
22. Görg, S., Bergmann, R.: Social workflowsvision and potential study. Information Systems **50**, 1 – 19 (2015). DOI https://doi.org/10.1016/j.is.2014.12.007. URL http://www.sciencedirect.com/science/article/pii/S030643791400194X
23. Grice, H.P.: Logic and conversation. In: P. Cole, J.L. Morgan (eds.) Syntax and Semantics: Vol. 3: Speech Acts, pp. 41–58. Academic Press, New York (1975)

24. Hernández, N., Ocaña, M., Alonso, J.M., Kim, E.: Continuous space estimation: Increasing wifi-based indoor localization resolution without increasing the site-survey effort. Sensors **17**(1), 147–170 (2017). DOI 10.3390/s17010147. URL http://dx.doi.org/10.3390/s17010147
25. InVerbis Analytics: https://processmining.inverbisanalytics.com/. [Online; accessed in January 2021]
26. Law, A.S., et al.: A comparison of graphical and textual presentations of time series data to support medical decision making in the neonatal intensive care unit. Journal of Clinical Monitoring and Computing **19**(3), 183–194 (2005)
27. van der Lee, C., Gatt, A., van Miltenburg, E., Krahmer, E.: Human evaluation of automatically generated text: Current trends and best practice guidelines. Computer Speech and Language **67**, 101–151 (2020). DOI 10.1016/j.csl.2020.101151
28. Leno, V., Dumas, M., Maggi, F., La Rosa, M.: Multi-perspective process model discovery for robotic process automation. In: CEUR Workshop Proceedings, vol. 2114, pp. 37–45 (2018)
29. Leopold, H., et al.: Supporting process model validation through natural language generation. IEEE Trans. Soft. Eng. **40**(8), 818–840 (2014)
30. Likert, R.: A technique for the measurement of attitudes. Archives of Psychology **140**, 1–55 (1932)
31. Mannhardt, F.: Multi-perspective process mining. In: BPM (Dissertation/Demos/Industry), pp. 41–45 (2018)
32. MATLAB in the Cloud: https://uk.mathworks.com/solutions/cloud.html. [Online; accessed in January 2021]
33. Ocaña, M., Llamazares, Á., Revenga, P.A., García-Garrido, M.A., Hernández, N., Álvarez, P., Fabra, J., Chapela-Campa, D., Mucientes, M., Lama, M., Bugarín, A., Alonso, J.M.: Estimation of customer activity patterns in open malls by means of combining localization and process mining techniques. In: Advances in Physical Agents II, pp. 30–43. Springer International Publishing, Cham (2021)
34. Ottensooser, A., Fekete, A., Reijers, H.A., Mendling, J., Menictas, C.: Making sense of business process descriptions: An experimental comparison of graphical and textual notations. Journal of Systems and Software **85**(3), 596–606 (2012)
35. Petre, M.: Why looking isn't always seeing: Readership skills and graphical programming. Commun. ACM **38**, 33–44 (1995)
36. RabbitMQ: https://www.rabbitmq.com/. [Online; accessed in January 2021]
37. Ramos-Soto, A., et al.: On the role of linguistic descriptions of data in the building of natural language generation systems. Fuzzy Sets and Systems **285**, 31–51 (2016)
38. Ramos-Soto, A., Bugarín, A., Barro, S.: Fuzzy sets across the natural language generation pipeline. Progress in Artificial Intelligence **5**(4), 261–276 (2016). DOI 10.1007/s13748-016-0097-x
39. Ramos-Soto, A., Bugarín, A., Barro, S., Taboada, J.: Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data. IEEE Transactions on Fuzzy Systems **1**(23), 44–57 (2015). DOI 10.1109/TFUZZ.2014.2328011
40. Reiter, E.: An architecture for Data-to-Text systems. In: 9th European Workshop on NLG (2007), pp. 97–104 (2007)
41. Reiter, E., Dale, R.: Building Natural Language Generation Systems. Studies in Natural Language Processing. Cambridge University Press (2000). DOI 10.1017/CBO9780511519857
42. Sànchez-Ferreres, J., Burattin, A., Carmona, J., Montali, M., Padró, L.: Formal reasoning on natural language descriptions of processes. In: T.T. Hildebrandt, B.F. van Dongen, M. Röglinger, J. Mendling (eds.) Business Process Management - 17th International Conference, BPM 2019, Vienna, Austria, September 1-6, 2019, Proceedings, *Lecture Notes in Computer Science*, vol. 11675, pp. 86–101. Springer (2019). DOI 10.1007/978-3-030-26619-6_8
43. Sànchez-Ferreres, J., Carmona, J., Padró, L.: Aligning textual and graphical descriptions of processes through ilp techniques. In: E. Dubois, K. Pohl (eds.) Advanced Information Systems Engineering, pp. 413–427. Springer International Publishing, Cham (2017)
44. Van Deemter, K., Krahmer, E., Theune, M.: Real versus template-based natural language generation: A false opposition? Comput. Linguist. **31**(1), 1524 (2005). DOI 10.1162/0891201053630291. URL https://doi.org/10.1162/0891201053630291

45. Vázquez-Barreiros, B., Mucientes, M., Lama, M.: Prodigen: Mining complete, precise and minimal structure process models with a genetic algorithm. Inf. Sci. **294**, 315–333 (2015). DOI 10.1016/j.ins.2014.09.057. URL https://doi.org/10.1016/j.ins.2014.09.057