Computers in Human Behavior 30 (2014) 583–594

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Understanding users' behavior with software operation data mining

Stella Pachidi [a,b,*], Marco Spruit [a], Inge van de Weerd [b]

[a] Department of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
[b] Information, Logistics and Innovation Department, VU University Amsterdam, De Boelelaan 1105, 1081HV Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Available online 21 August 2013

Keywords:
Software usage
User behavior
Software operation knowledge
Software analytics
Log data
Data mining

ABSTRACT

Software usage concerns knowledge about how end-users use the software in the field, and how the software itself responds to their actions. In this paper, we present the Usage Mining Method to guide the analysis of data collected during software operation, in order to extract knowledge about how a software product is used by the end-users. Our method suggests three analysis tasks which employ data mining techniques for extracting usage knowledge from software operation data: users profiling, clickstream analysis and classification analysis. The Usage Mining Method was evaluated through a prototype that was executed in the case of Exact Online, the main online financial management application in the Netherlands. The evaluation confirmed the supportive role of the Usage Mining Method in software product management and development processes, as well as the applicability of the suggested data mining algorithms to carry out the usage analysis tasks.

1. Introduction

Software usage concerns the utilization of a software product by the end-users. Software usage data may be collected while the end-users are using the software in the field (El-Ramly & Stroulia, 2004). Simmons (2006) points out the possibility to extract system requirements from usage, rendering the beneficial role of user experience in product innovation and differentiation. Software usage knowledge includes the awareness of how end-users use the software in the field, and how the software itself responds to their actions (Van der Schuur, Jansen, & Brinkkemper, 2010).

By tracking software usage, we can monitor which applications are most often used, which features are underutilized, and which functionalities could be expanded (Junco, 2013). This information could for example be used to highlight changes in the requirements engineering process. We may also gain insights on how users browse themselves through the user interface in order to perform an operation, with the goal to improve software usability or to reengineer processes. Furthermore, by observing the usage behavior of different customer profiles, the software vendor can implement more directed marketing or customized licensing (Germanakos, Tsianos, Lekkas, Mourlas, & Samaras, 2008; van der Schuur et al., 2010). Improved customer satisfaction, and consequently customer retention and increase in sales, are some of the business advantages that could be gained through an automated usage analysis, based on real execution data.

Software usage knowledge may be extracted from *software operation data*, i.e. data that are collected during software operation in the field (van der Schuur et al., 2010). A noticeable amount of research has already been performed in the process of recording software operation data (Bowring, Orso, & Harrold, 2002; Nusayr & Cook, 2009). In practice, most vendors tend to handle the acquired data manually, or use general statistics and simple visualization techniques (Kristjansson & Van der Schuur, 2009). However, such analysis cannot yield interesting patterns that are hidden in large datasets (Kantardzic, 2002).

On the other hand, a lot of development has been seen in the web usage mining field (Cooley, Mobasher, & Srivastava, 1997). Although many lessons can be learned from there, the approach for analyzing web usage by website visitors has significant differences, compared to analyzing how software products are used by the users. The techniques that are used in web usage mining (and other related domains) need to be revised for their application in mining usage on software operation data.

While usage knowledge is highly important for making good software products, the rise of cloud computing and Software-as-a-Service (SaaS) applications (Park & Ryoo, 2013) creates an opportunity to mine the easily acquired data. Even though there are algorithms for doing such data analysis, they are hardly ever used for analyzing software usage. Following a meta-algorithmic approach, we will try to answer the research question:

How should we inspect software operation data, in order to gain knowledge about how the software is used by the end-users?

* Corresponding author at: Information, Logistics and Innovation Department, VU University Amsterdam, De Boelelaan 1105, 1081HV Amsterdam, The Netherlands. Tel.: +31 644478898.

E-mail addresses: s.pachidi@vu.nl (S. Pachidi), m.r.spruit@uu.nl (M. Spruit), i.vande.weerd@vu.nl (I. van de Weerd).

0747-5632/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.chb.2013.07.049

This research suggests how data mining techniques can be integrated to analyze software operation data in a uniform and automated way. Hence, it contributes to the domain of software usage analysis as well as to the software operation knowledge and its use in software product management, development and maintenance processes (Van der Schuur et al., 2010). From a practical perspective, the method that we suggest for usage mining constitutes a reference process model that can be followed by software vendors, to analyze how their customers use their products.

The remainder of this paper has the following structure: In Section 2 we review the research that has been performed on the area of extracting usage knowledge from the system utilization. We shortly present our research design in Section 3. In Section 4 we present the method that has been constructed to extract usage knowledge. In Section 5 we describe the usage knowledge subjects that we suggest to extract, and the variables that should be inspected in software operation data, in order to derive conclusions about how software operates in the field. Section 6 describes the data mining techniques that are suggested for mining software usage knowledge. In Section 7 we present the prototype that was constructed as an instantiation of the usage mining method. We evaluate the two artifacts in a case study in Section 8. Finally, in Section 9 we discuss the insights from this research and provide some general conclusions.

2. Related work

As far as specific research on software usage analysis is concerned, extraction of in-the-field usage knowledge remains an area that needs a lot of enrichment. Data analysis techniques have been previously applied to this field: for software reengineering purposes (El-Ramly, Stroulia, & Samir, 2009; Lefngwell & Widrig, 2003), for program comprehension (Zaidman, Calders, Demeyer, & Paredaens, 2005), for re-documentation of use cases (Smit, Stroulia, & Wong, 2008), or for user interface learning agents (Ruvini & Dony, 2001). However, these approaches are not directly related to analyzing how the end-users are utilizing the software in the field. Also, they do not provide any holistic approach to the various usage knowledge types (e.g. user profiles or most frequent navigation paths). Some of them are very old, so they do not use state of the art data mining techniques.

Several techniques have been developed for deriving models based on analysis of log files (Petruch, Tamm, & Stantchev, 2012). For example, analyzing the audit trails through sequence analysis techniques can prove to be quite useful for evaluators who are curious to compare the designers' expectations of use with the actual usage patterns followed by the users (Judd & Kennedy, 2004). This approach is similar to the field of Process Mining (Van der Aalst & Weijters, 2004), which involves analysis of event logs with the goal to monitor and/or redesign operational business processes that take place in an information system (Maruster & van Beest, 2009). Process Mining has also been applied on web services workflows mining (Dustdar & Gombotz, 2007).

Practical examples that include usage analysis of logged events can be found in literature (Lin & Tsai, 2011). Transaction logs analysis techniques are used in the usage analysis of a digital library (Jones, Cunningham, & McNab, 1998). Shen, Fitzhenry, and Dietterich (2009) use a subgraph mining algorithm, in order to partially automate the user's workflows or to create to-do lists, in a desktop assistant application. Sartipi and Safyallah (2009) developed a data mining algorithm for sequential pattern discovery on traces that are generated from the execution of task scenarios.

A closely related area to software usage analysis is web usage mining, one of the subfields in web mining. Web usage mining is the process of automatically discovering and analyzing behavioral patterns and users profiles in clickstream and other associated data, which are generated or collected when users interact with web resources found on one or more websites (Liu, 2006). The most common pattern discovery and analysis tasks include: session and visitor analysis (Liu, 2006), visitor segmentation and profiling (Xie & Phoha, 2001), association analysis (Meo, Lanzi, Matera, & Esposito, 2006), navigation analysis or path analysis (Cooley, Mobasher, & Srivastava, 1999), and prediction based on web user transactions (Liu, 2006).

3. Research design

The users' shift to cloud computing applications (Park & Ryoo, 2013) creates the opportunity for software vendors to automatically collect vast amounts of usage data. Although several algorithms have been developed to analyze the behavior of website visitors, they are hardly ever used in the software products domain. This research aims to follow a meta-algorithmic approach, by incorporating the state-of-the-art data mining techniques in a method. Our goal is to show how the appropriate technique can be used for analyzing each aspect of software users' behavior.

In Fig. 1 we display a diagram of our research design. We follow the *design science research* (*DSR*) approach (Hevner, March, Jinsoo, & Ram, 2004), as we develop a method and a prototype for software usage mining. We follow the General Design Cycle (Vaishnavi & Kuechler, 2007), which includes the phases: problem awareness, suggestion of a tentative design, development of the artifact, evaluation, and conclusion.

We construct a method for usage mining using the Method Engineering approach provided by van de Weerd and Brinkkemper (2008). To evaluate the effectiveness and applicability of the method, we perform a case study in an international software company, for analyzing the usage of an online financial application by trial customers.

In order to structure our data mining research, but also to assemble our Usage Mining Method, we follow the CRISP-DM Reference Model (Chapman et al., 2000), which includes six phases of data mining activities: business understanding, data understanding, modeling, evaluation and deployment. In Section 4 we show how these activities were incorporated in the method.

For the evaluation of this research we use Case Study Research (Runeson & Höst, 2009) and follow a positivist approach. The case study takes place in the context of Exact Online, an online financial management application, and consists of four phases: (1) design and preparation, (2) conducting, (3) analyzing and (4) reporting. In Section 8 we describe these phases in detail.

4. Usage Mining Method

In this section we present the first design artifact that we constructed in this research. The Usage Mining Method suggests an ordered set of activities that should be followed to extract relevant usage knowledge from software operation data.

In order to provide guidance in analyzing software product users' usage behavior, we propose the Usage Mining Method (Fig. 2). The method has been constructed with the Method Engineering approach, provided by van de Weerd and Brinkkemper (2008). The method is designed for mining usage, user and corporate data of software-as-a-service applications, which are collected in a central point on the software vendor's side.

Fig. 2 includes an overview of the method's activities and sub-activities. The activities that are connected through an arrow are sequential, i.e. they need to be carried out in a pre-defined order, for the reason that the outcomes of the former activities are

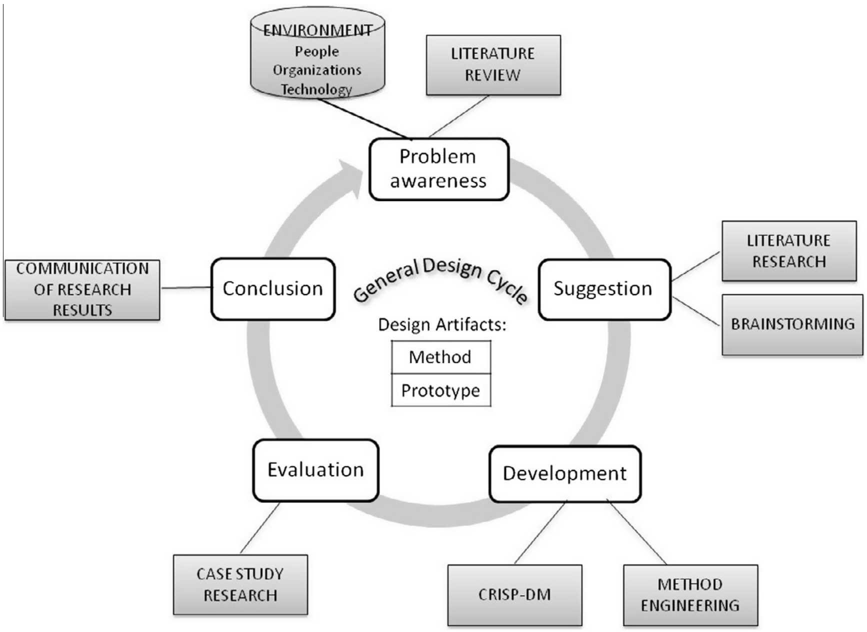

Fig. 1. Diagram of our research design.

essential for the execution of the latter. Activities drawn in parallel may be executed concurrently.

The method starts with the activity of *Data Understanding*, which includes the selection of variables that need to be logged, the logging procedure, the description of the data quantity and format, and the evaluation of data quality (in terms of completeness, inconsistencies, and duplicate data). The subsequent activity is called *Data Preparation*, and includes all steps related to data preprocessing (data selection, data transformation, data cleaning, data construction, data integration), in order to produce the final dataset that can be used for the analysis. Then we perform *Exploratory Analysis*, in which we perform statistical analysis on the data set to produce general measures and figures that describe the users' behavior and are used as input in the data mining tasks. The three main analysis tasks correspond to the activities: *Classification Analysis*, in which we build a classification tree and then evaluate the model with cross-validation; *Users Profiling*, in which we build several clustering models by running different clustering algorithms, and then we validate the results and select the clustering with optimal scores on the validation measures; and *Clickstream Analysis*, in which we build Markov chains and mine sequential patterns, to select interesting usage paths. The final activity of the method is the *Evaluation*, which consists in evaluating the results from the activities of classification analysis, users profiling and clickstream analysis, in terms of business success criteria.

5. Software usage knowledge

In this section we suggest what types of knowledge should be extracted from software operation data to gain insights about how the end-users are using a software product. Subsequently, we present the fundamental variables that should be inspected during software operation, in order to gather the data that are necessary to analyze usage.

Based on our findings from our literature research in the domains of usage analysis in software systems (El-Ramly et al., 2009; Simmons, 2006) and web usage mining (Liu, 2006; Srivastava, C-ooley, Deshpande, & Tan, 2000) as well as the insights from our case study, we suggest four categories of usage knowledge that could be derived from software operation data:

- *Statistical summary of sessions and users' behavior*: A statistical summary of the usage related data helps us infer conclusions about the general behavior of all users and their sessions. Examples: most frequently used functionalities, average session duration.
- *Factors that influence the customers' decisions*: Decisions such as renewing or upgrading the license may be predicted if we analyze the factors that influence the customers, based on how they have been using the product (Liu, 2006).
- *Users Profiles*: Extracting users profiles based on their usage behavior (Srivastava et al., 2000) consists in creating segments of users who use different sets of functionalities and visit different sets of pages.
- *The Most Frequent Navigation Paths*: Knowing the most frequent navigation paths helps us analyze the usage and usability of the software product. Furthermore, by analyzing inter-session patterns of users we can make predictions about their future actions (Lee, Podlaseck, Schonberg, & Hoch, 2001).

In order to extract the aforementioned knowledge categories from software operation data, we can apply various data analysis techniques, which we study in Section 6. However, it is also important to first specify what variables should be inspected in software operation data, in order to draw conclusions about usage. Based on the usage model of Simmons (2006) and the data types used in web mining research (Srivastava et al., 2000), we distinguish three *categories of variables*:

- *Usage data*: This category includes 18 operation details variables, recorded during the software usage, which describe the end-users actions on the software product. We want to log: who is using the product (customer id, user id, IP address);

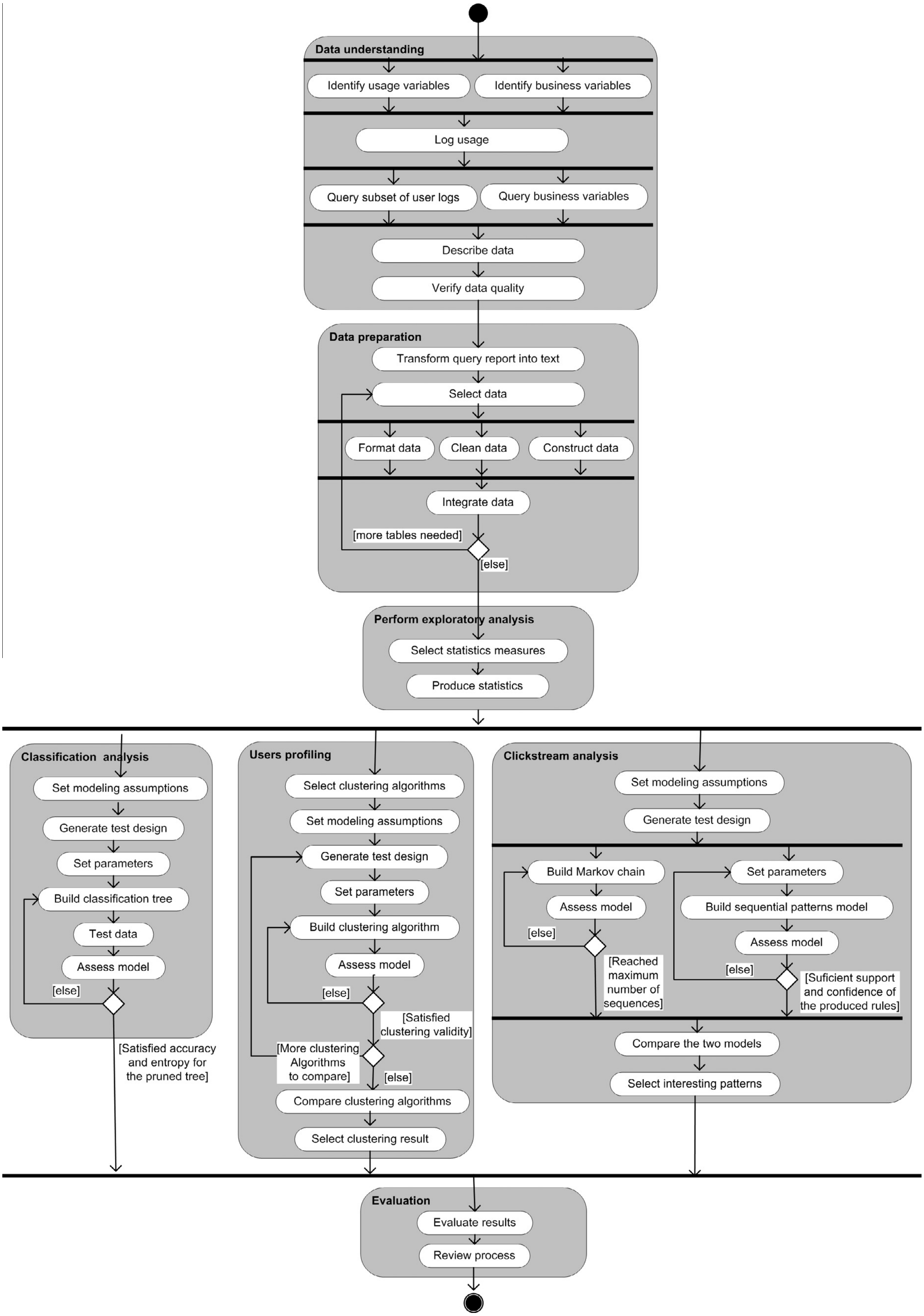

Fig. 2. Process diagram of the Usage Mining Method.

where the application is being hosted (web server, database); what the end user does (application, page, method, function, button that is accessed, action that is performed); when the user performs the operation (data and time, session id); how long it takes to complete an operation (duration, query duration); and other operation details (errors, background tasks, number of records loaded).
- *User data*: This category refers to variables that describe the profile of each user, such as demographic information. The 10 most common variables that are used include: age of the end-user; level of education; region, city, area of residence; the type of license that the user has purchased; the size of data records stored in the system's database for the specific user; user ratings on various objects; information on the user settings such as type of browser used when accessing the application and operating system; date that the user account was created.
- *Corporate data*: This category is applicable to corporate customers, and includes variables that provide information about the customer organization, such as: in which domain the organization operates (e.g. construction, logistics, government, education, etc.); head-count of the organization; number of administrations that the organization consists of; number of users who use the software product; license information such as type of license and data of purchase.

In this section different knowledge types related to software usage and different variables that may be inspected in software operation data were presented. In the next section we will look into the data mining techniques that may be used for the analysis.

6. Usage mining tasks and data mining techniques

In this section, we are going to suggest which data mining techniques could be performed on the software operation data, in order to analyze the software usage. More specifically, in order to produce the various usage knowledge types presented in the previous section, we suggest the following usage analysis tasks:

1. *Classification Analysis*, to understand the factors which influence the decisions that customers take, in the context of the software product utilization.
2. *Users Profiling*, i.e. segmentation of the user records (sessions or transactions) or the users, in order to extract profiles, that describe different navigation behaviors, or different groups of users with similar interests, respectively.
3. *Clickstream Analysis*, to extract different usage scenarios, represented by the most frequently followed navigation paths.

In the following sub-sections we will present the main data mining techniques that can be used for each of the remaining analysis tasks. More emphasis is given on the techniques that we implemented in our prototype. These techniques were selected having as criteria their implementation in R (advising the available packages in R library as of July 2011), their ease of use (by experimenting with the algorithms in the prototype construction and by communicating with data mining experts from the R-help mailing list), and their prior use in related work (based on the literature review in the domains described in Section 2). Table 1 enlists a rating of the presented techniques based on these criteria.

6.1. Classification analysis

This task is performed with the further purpose to improve the conversion rates (Lee et al., 2001) related to the context of the software product utilization. We aim to understand the factors that influence the decisions that customers take when they use a software product, such as whether they will convert from the trial to the regular version of the product or whether they will update their license. By extracting these factors, we can predict their next decisions, and gather feedback to improve the product design and customization, or the placement of advertisements in a web application.

This task requires all types of variables that we presented in Section 5. User data and corporate data are directly related to each customer who uses the software product. However, as far as the usage data are concerned, *aggregate variables* need to be created, to describe the usage data, so that in the end we have one record per customer. These aggregate variables are produced during the exploratory analysis task (Giudici, 2003), and they may be for example the frequency between logins, the average session length of the user, or the total time spent on utilizing the product.

Three different techniques can be applied in this task:

- *Logistic Regression Models*, through which we can predict the expected outcome value for an object, based on the related scores on the predictor variables (Field, 2009). A *binary logistic regression* model is suggested, in which the response variable describes the outcome of the customer's decision, e.g. whether the customer will convert or not. The set of predictor variables consists of the several usage, user and corporate attributes.
- *Classification Tree Models*, a popular data mining technique for predicting the outcome class of an object which offers easy interpretation, automatic selection of the relevant attributes, and the ability to handle both numeric and categorical attributes (Breiman, Friedman, Olshen, & Stone, 1984). The structure of a *classification tree* (also referred to as decision tree) looks like a flowchart, in which each internal node (i.e. non-leaf node) represents a test on an attribute, each branch denotes a test outcome, and each leaf-node corresponds to a class label (Han, 2005). To construct a classification tree (Fayyad & Irani, 1992), we use

Table 1
Data mining techniques for usage analysis.

Analysis task	Data mining technique	Implementation possibilities in R	Ease of use	Prior use in related work
Classification analysis	Logistic regression models	***	***	*
	Classification tree models	***	**	***
	Multilayer perceptron models	*	*	*
Users profiling	Cluster analysis	***	***	***
	Kohonen maps	**	**	*
Clickstream analysis	Sequential pattern mining	*	**	***
	Probabilistic expert systems	*	*	*
	Markov chains	**	***	***

historical data that have pre-defined classes and other attributes. The tree building algorithm calculates *tests/splits* which predict the available objects' classes with the best accuracy. In the case of analyzing the factors that influence the users' decisions, the classes represent the outcomes of the decision (yes/no or 1/0). We get to have as a sample the total number of customers, with attributes the aggregate usage variables, the user and the corporate variables. From this set of cases, we could construct (and cross validate) a classification tree that best describes our data. We can then study the splitting variables, which indicate the factors that are related to the decision of the user. But also, we can use this tree as future reference to predict the outcome of decision of other users.

- *Multilayer Perceptron Models*, which can be used for the task of credit scoring (Giudici, 2003). A multilayer perceptron (MLP) is an artificial neural network model and is represented by a directed graph, which comprises multiple interconnected layers of nodes (Haykin, 1998). In our case we can choose a perceptron model to classify the customers into two groups (e.g. decide to convert and not convert) according to the values of the usage, user and corporate variables, and then make predictions.

6.2. Users profiling

In the task of users profiling, we wish to create profiles of the users, based on their navigation behavior, which group the users according to their similar interests. The retrieval of users profiles can increase the marketing intelligence and help attract new customers more effectively, help the vendors offer solutions that are effectively targeted to each user, and in the end retain the old customers in a more adept way. Another interesting advantage that the navigation profiles can give us is the observation of a user's navigation behavior over time, thus we can observe the evolution of individual customers but also the evolution of the type of customers who use the software product over time.

In order to perform this task, we need to have usage data, i.e. user records which represent the operations that users perform on the software product, and have as attributes the operation details (such as which application was accessed, when, and by which user). In order to prepare the dataset for this task, we need to perform exploratory analysis on these data, and generate aggregate variables, which describe the navigational behavior of each user. The goal is to create a matrix, containing one record per user, with aggregate attributes that describe how often, or for how long, the specific user used each function of the product. When a product has many functions, it is advised to organize them in homogeneous categories, which reflect their logical meaning and functionality (Giudici, 2003).

We suggest two techniques for performing this task:

- *Cluster Analysis: Clustering* is the process of organizing objects into a number of groups, in such a way that the objects within the same group are similar with each other and are dissimilar with objects that are members of other groups (Kaufman & Rousseeuw, 2008). Each object may be described by a *data point* in a multidimensional space, in which the dimensions correspond to the variables that describe the data (Jain, Murty, & Flynn, 1999). A cluster is viewed as an aggregation of points that are positioned closely to each other. The main goal in clustering is to minimize the within-groups distances and maximize the between-groups distances (Han, 2005).

In the context of usage analysis, our data set is a matrix with m rows that correspond to the different users and n columns that correspond to the categories of the product's pages/functions, etc. Each cell $[i,j]$ in the matrix represents the frequency of uses or the total time of using the pages of the category j by the user i. Then we have a n-dimensional space, in which each dimension corresponds to a category of pages and each user is represented by a data point. Thus, the objective of clustering is to group together the users who have similar viewing with each other and dissimilar with users from other clusters. A simplistic example drawn from our case study is reflected in Fig. 3. Each cluster represents a usage profile, which could be represented by the center of the corresponding cluster; the center indicates the navigational behavior of the users who share the same profile.

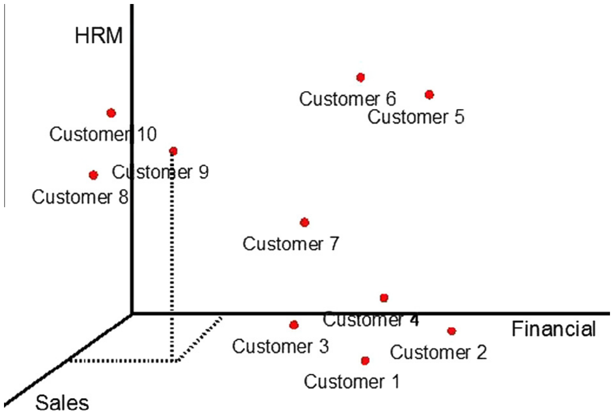

Fig. 3. Simplistic example of clustering the users in an online accounting solution.

Several clustering techniques have been developed in the data mining domain. We have tested and suggest hierarchical clustering (Tan, Steinbach, & Kumar, 2005) and specifically Ward's method (Hastie, Tibshirani, & Friedman, 2009), as well as partitional clustering (Jain et al., 1999) and specifically: the K-Means algorithm (Everitt, Landau, & Leese, 2001), the K-Medoids algorithm (Everitt et al., 2001) and fuzzy clustering (Xie & Beni, 1991).

- *Kohonen Maps: Self-organizing maps* (*SOMs*) or *Kohonen networks* (Kohonen, 2001) are a typical implementation of *unsupervised neural networks* that can be used in the context of descriptive data mining, when we want to cluster a set of observations into homogeneous groups. A Kohonen network has one input layer and one output layer. Our objective is to divide n observations (each of which is a p-dimensional vector that contains numeric and categorical attributes) to a pre-defined number k of groups (clusters). The multivariate n observations correspond to n input neurons of the network, while the k output nodes are described by discrete values 1, ..., k which indicate the different groups. The objective of a Kohonen map is to map each p-dimensional input observation to an output space, which is represented by the spatial grid of output neurons (Kohonen, 2001). The winning output for each input observation is the closest cluster based on a predetermined distance function. However, the assignment is done in such a way that the neighboring relations between the clusters are preserved to some degree.

Going back to the usage analysis, our data set consists of n observations that correspond to the different users, and each observation is a p-dimensional vector with the frequencies that the corresponding user has visited the pages of the p categories. The n user records will be the n input neurons in the Kohonen Network (Giudici, 2003). We need to select the desired number k of usage profiles that we want to extract. We also need to select the number of rows a and the number of columns b in the grid space that will characterize the map, so that $a \times b = k$. Then the objective of the Kohonen map is to map each of the n user records to one of the k output neurons, which are positioned in the $a \times b$ map. In order to understand the cluster configurations, we can inspect the center of each cluster, which will be indicative of the navigation behavior of the users in the same cluster.

6.3. Clickstream analysis

While the end-users are using a SaaS application, the logging procedure keeps track of all their actions, which are centrally stored in a log file on a server (Stieger & Reips, 2010). We can therefore capture the navigational paths that each end-user follows when using the product. The analysis of these navigational paths is often called *clickstream analysis* (Giudici, 2003), and can be used to extract different usage scenarios, analyze the usability of the software, and predict the next actions of the user.

The dataset consists of usage data, which describe the users' actions. More specifically, it is a long matrix in which each row corresponds to a page-view (i.e. any kind of user action such as the view of a certain functionality) and each column corresponds to a specific usage variable (timestamp of the action, user id, the page viewed). The data need to be organized in *sessions*, each of which describes the succession of pages viewed by a user during a limited time period (e.g. from the moment he opens until he turns off the application or from the moment he logs in until he logs out). The simplest form of clickstream analysis is to remove the user information after deriving the sessions, and hence try to find similarities between sessions. The more complex form is to organize the data into sessions per user, and look for common patterns, not only between page views, but also between successive logins.

We are suggesting three techniques for analyzing navigational patterns:

- *Sequential Pattern Mining*: Sequences of events that describe the behavior and actions of users or systems can be generated often in several domains. Sequential pattern mining is a technique based on frequent pattern mining, but here the rules that we try to mine have the format $A \rightarrow B$, meaning that if episode A occurs, then episode B is likely to occur subsequently.

As far as the usage data are concerned, the transactions represent user sessions, i.e. trails of the users' actions. A simple example of a rule that we could extract could be *catalogue* → *product* which means that when the user views the "catalogue" he will then view the "product", or {*catalogue* → *product*} → *addcart* which means that if the user views the "catalogue" and then the "product" page, he will likely view the "addcart" page next.

- *Probabilistic Expert Systems:* Probabilistic expert systems are graphical networks that try to process the encoded knowledge from a knowledge base, in order to model the uncertainty and decisions in large complex domains (Cowell, Dawid, Lauritzen, & Spiegelhalter, 2007). The strategy of probabilistic expert systems to build up a global statistical model is based on subsequent local factorizations. Here, a rule $A \rightarrow B$ means that the page B will be visited only if the page A has been visited. If the support of this rule is higher than a predefined threshold, it is established as a valid rule.
- *Markov Chains:* A *Markov chain* is a mathematical system that goes through transitions from one state to another based on probabilities in a chainlike manner (Grinstead & Snell, 2006). We call as *order* of the Markov model the number of prior events that contribute to the prediction of a future event. In the simple case of a first-order Markov chain, what happens at time t depends only on the event that happened at time $t - 1$ (Giudici, 2003). Markov models of higher order predict with higher accuracy, but result in lower coverage (recall) and very high computational complexity (Liu, 2006).

The dataset that is considered for the clickstream analysis consists of rows, which represent the actions of users, roughly called pageviews, with columns the details of the operation case (timestamp, page viewed, action, user id, etc.), organized in sessions, whereas each session constitutes a sequence of pageviews. We could consider each pageview as a possible *state* and create Markov chains to model the usage (navigational) behavior of the end-users. We can then produce navigational paths, which are represented by paths that connect nodes through the most likely transitions.

Also, we can obtain useful information such as the likelihood of moving from one page to another; or the most frequent entry/exit point for a session.

In this section, we presented how we can use data mining techniques to analyze data related to the end-users behavior, actions and decisions. For the classification analysis three techniques were presented: logistic regression models, classification tree models, multilayer perceptron models. Two techniques were suggested for users profiling: cluster analysis and Kohonen maps. Clickstream analysis may be performed through three techniques: sequential pattern mining, probabilistic expert systems and Markov chains.

7. A prototype for usage mining

The Usage Mining Method presented in Section 4 is instantiated in a prototype, which we developed in R (R Development Core Team, 2008) and implements the method's activities. The prototype can be used to analyze the software usage of SaaS products with embedded logging procedures that record the operation data. The prototype has the format of an *R script*, which performs successively the activities of Data Preparation, Exploratory Analysis, Classification Analysis, Users Profiling and Clickstream Analysis.

For the *classification* task, we used the *R* packager *part*, which includes programs that build regression and classification models (Therneau & Atkinson, 1997), to grow a customer classification. The case of a new customer may be inserted in the classification tree, and, through several tests, it may be assigned with the class label 1 (the customer will convert) or 0 (the customer will not convert).

Since the result of *clustering* analysis is quite dependent to the method used and to the selected number of clusters, in the prototype we implement several methods; based on a set of evaluation criteria, the analyst can select the clustering result that is more appropriate each time. More specifically, the prototype implements: hierarchical agglomerative clustering with Ward's method (R function hclust); partitional clustering with the K-means (R function kmeans) and the K-medoids algorithm (R function pamk); fuzzy clustering with the C-Means algorithm (R function cmeans) and Kohonen maps (functions somgrid and som from the R package kohonen).

For the *click stream analysis* task, the prototype creates the probability transition matrix, in which the element (i,j) denotes the probability of viewing page j after viewing page i. The probability transition matrix can then be used to build first-order Markov chains that model the users' navigational patterns. Other interesting conclusions may be inferred from the matrix, such as the most common entry and exit points of a session.

The output of the prototype execution is exported in images and documents of CSV format, in a specified output folder, so that they can be inspected at any time. In order to use the prototype with the logs of a specific product, the analyst has to modify some fields in the R script (guided by comments on the code), and then select the lines and execute.

8. Case study

Following the Design Science Research approach, we just presented the two design artifacts that we constructed in this research: the Usage Mining Method and the prototype developed in R. In order to evaluate the two artifacts, we performed a case study in a Dutch software company, to implement the Usage Mining Method and run the prototype in the context of a real software product. This section comprises the design of the case study, as well as the execution and interpretation of the results.

8.1. Case study design

8.1.1. Case study objectives

The case study was performed to evaluate the Usage Mining Method and the Usage Mining Prototype, in response to our main research question: how we should inspect software operation data, in order to gain knowledge about how the software is used by the end-users.

8.1.2. Case selection

In order to evaluate the Usage Mining Method and Prototype, we performed a case study in Exact,[1] a Dutch software company that serves small to medium enterprises with information technology, by delivering business software solutions. Our case study was performed in the context of *Exact Online*, an internet-based accounting solution which constitutes one of the main software products of the company. Exact Online is a *Software-as-a-Service* application, with over 10,000 customers and more than 3500 users per day.

Exact Online logs performance and usage data through a log-generating code that is incorporated in the system layer of the application. Every time the application is used by an end-user, several variables are stored in log tables in the administration database, on the server's side. The information that is logged contains: usage data (user id, application used, date/time, action taken, etc.), performance data (how long it took to process a query, how long it took to load an application, etc.), quality data (e.g. errors that appeared during usage) and other useful information such as which background tasks were running during usage. All logs are stored in the database of Exact Online for the period of 90 days. Currently, such log information is analyzed by manual inspection, through the use of standard statistics. However, this kind of analysis turns out to be very difficult, error-prone and time consuming. Soon, as users of Exact Online continue to increase significantly, it will even become more difficult. At the same time, manual inspection fails to identify complex patterns between these data.

Exact offers the possibility to anyone interested in trying out the functionalities of Exact Online, to register for a 30-day free trial. After registering for the trial, the customers can use the functionalities of the product and decide whether to buy a subscription for one of the packages offered.

The software operation data of Exact Online could provide us with valuable information on the first experiences of the end-users with the product, such as how they browse the pages in order to discover the functionalities. This kind of information is particularly useful: to understand how the usability of the product could be improved; to understand what factors influence users to buy a license; but also to segment customers according to their usage profiles, and help the marketing department target them more effectively (Okazaki, 2007). We applied the suggested Usage Mining Method and used our prototype to extract such kind of usage knowledge from the operation of the trial version of Exact Online. We discuss the process and results from this analysis in the following paragraphs.

8.1.3. Data collection procedure

Data was collected through documentation (reports, software manuals, software architecture specifications, memos, etc.), direct observations (to understand what the developers need to know about the usage of their software), exploratory interviews (with product managers, software engineers, research engineers and a data mining expert) and participation in experts meetings.

As far as the collection of the data needed for the analysis tasks is concerned, our main dataset consisted of usage data logged during the use of the trial version of Exact Online over a period of 4 months (March–July 2010). This dataset consisted of approximately 440,000 rows with 12 attributes, and corresponded to 908 customers. Additional data were collected regarding the accounts, software quality/errors, access to help documentation, and statistics on the size of each customer account. Demographic and corporate information regarding the users was not provided for reasons of protecting the customers' privacy.

8.1.4. Analysis procedure

The analysis of qualitative data was performed incrementally with the data collection procedure. Coding and tabulation of the data helped identify what types of data would need to be queried (from the logs and other data sets). The configuration of the Usage Mining Prototype could then take place, as some pre-processing code had to be added to serve this analysis, but also several parameters had to be arranged in the analysis code.

Before performing the data mining tasks, we first performed an exploratory analysis, in order to get some general statistics on the customers of the trial Exact Online and their usage behavior. We calculated the conversion rate for the customers who were included in our logs, the average view time per page, and so on. The three main analysis tasks followed: classification analysis, users profiling and clickstream analysis. The results are provided in Section 8.2.

8.1.5. Validity

Attention was paid to all aspects of validity (Runeson & Höst, 2009): To ensure construct validity, we used multiple sources of evidence (triangulation): documentation, direct observations, exploratory interviews. Internal validity was threatened for the classification analysis task and for the users profiling task of the method, because we focused on the usage variables and omitted most demographic and corporate variables, as these were not provided for reasons of privacy. The results will be biased in these two cases. However, the two analysis tasks would be performed in a similar way if those types of variables would also be included, while their usefulness has been previously discussed in the literature – e.g. in the work of Srivastava et al. (2000). In order to ensure the external validity, we performed an extensive literature review; had our method reviewed by a peer researcher (data mining expert); and we repeated the analysis with data collected during a different time period (within-case examination). By developing and maintaining a detailed case study protocol and spending sufficient time with the case, we ensured that the procedures used

[1] http://www.exact.com/.

were well documented, and that they could be repeated again (reliability).

8.2. Case study results

In the following sub-sections we examine the analysis tasks that were performed on the usage data of the trial version of Exact Online. For each task we describe how the analysis was executed and how the data mining results were validated, while afterwards we provide an interpretation of the results.

8.2.1. Classification analysis

8.2.1.1. Analysis execution. As far as the classification analysis is concerned, we studied the variables that are related to the customers' decision to buy a license, after having used the trial version of Exact Online. In order to validate the classification results, we decided to split our dataset in a set of training data (650 user records) which will be used for growing the classification tree and a set of test data (258 records) the classes of which we will predict with the fitted model(s). We selected which variables would be included in the data set and ran the classification tree building function.

8.2.1.2. Validation of the data mining results. The resulting classification tree was pruned (Fig. 4) to avoid overfitting and contained two factors: number of page views, and total size of the stored records. Classifying the 258 test observations with the resulting tree was accurate for 85.7% of the cases.

8.2.1.3. Interpretation. The number of page views denotes the amount of activity of the end-users in the application. Naturally, the users who decided to convert seem to have spent sufficient time using the trial version. On the other hand, it is somewhat surprising that higher size of a customer's administration data, leads to a decision not to convert. Perhaps there were delays in the loading of these records, which demotivated the users.

Through the classification tree analysis, the stakeholders of Exact Online found some interesting insights on how usage was related to the customer's decision to convert or not. However, we would expect the results to have been more insightful if demographic and corporate data had also been included in the variables of the classification tree.

Table 2
Validation of the clustering results for the users of trial exact online.

	Hierarchical clust.	K-means	K-medoids	Fuzzy clust.	Kohonen maps
Number of clusters	5	5	2	5	10
Connectivity	182.19	85.04	49.22	317.83	151.71
Dunn index	0.003	0.025	0.028	0.001	0.021
Silhouette width	0.426	0.706	0.811	0.166	0.669

8.2.2. Users profiling

8.2.2.1. Analysis execution. In the users profiling task we grouped the users into clusters based on their navigational behavior in the trial version. We used all the aforementioned clustering algorithms. A test design was generated to run the clustering algorithm multiple times before selecting the correct number of clusters and stopping criterion. Then the clustering models were built.

8.2.2.2. Validation of the data mining results. Five clusters were extracted with the k-means, two clusters with the k-medoids, five clusters with the fuzzy c-means, and ten clusters with the Kohonen maps procedure. In order to validate the clustering results, we used the internal validation measures of connectivity, Silhouette width and Dunn index (Brock, Pihur, Datta, & Datta, 2008). The measurements can be viewed in Table 2. The k-medoids algorithm gave the optimal scores (lowest connectivity, highest Dunn index and Silhouette width) by giving two clusters of users: the ones with low usage frequency the ones with high usage in some categories.

8.2.2.3. Interpretation. The centers of the five clusters from the k-means algorithm can be viewed in Fig. 5, indicating the amount of usage in each category of pages. In all partitions shown in Table 2 we noticed the existence of a cluster (cluster 5 in Fig. 5) that contains most of the observations, indicating the profile of users who had very low usage. This is not unexpected, since we clustered users of the trial version, and the majority of them had used it very few times. Analyzing the customers who have bought the commercial version and use it regularly, would have provably given more distinct cluster profiles.

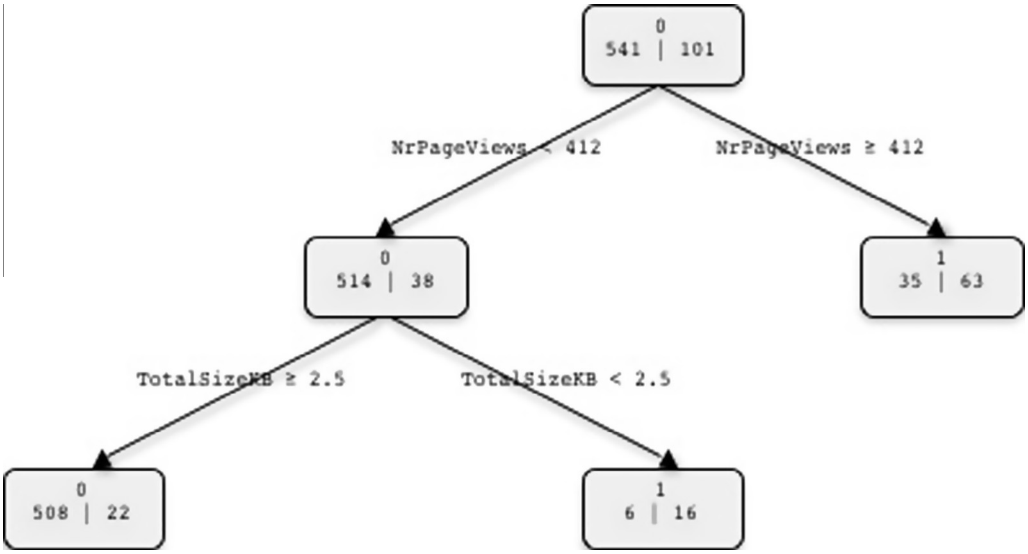

Fig. 4. The pruned classification tree, relating the number of page views and the size of the user's stored transactions to the conversion decision.

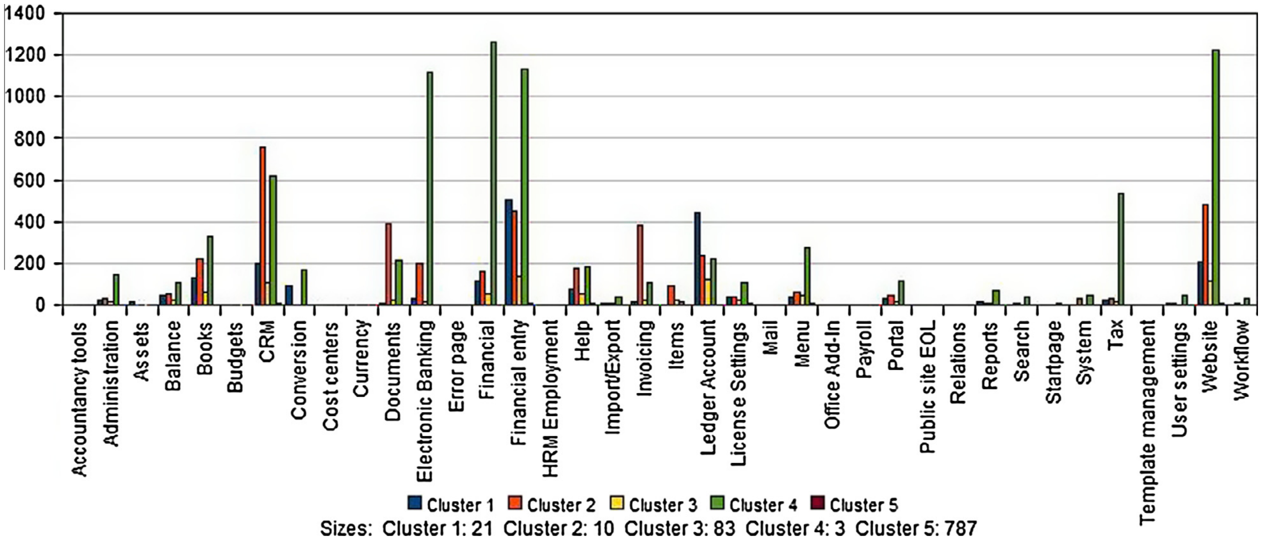

Fig. 5. Cluster centers for the k-means algorithm.

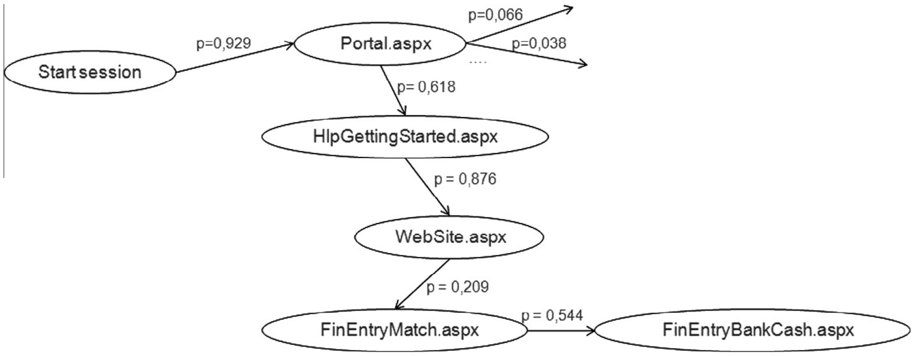

Fig. 6. Visualization of a Markov chain.

8.2.3. Clickstream analysis

8.2.3.1. Analysis execution. We performed Markov chains analysis to find common navigational patterns. A test design was generated, which instructed the creation of 10 different possible Markov chains and one chain with the most probable transitions, as well as the extraction of other interesting clickstream data, such as the most likely entry and exit points and the most likely transitions to/from a set of specific pages. An example of how a Markov chain could be visualized is exposed in Fig. 6.

8.2.3.2. Interpretation. From the Markov chains, the developers of Exact Online could gain insights about the navigational patterns that the users follow in the trial version and thus analyze the first experiences of the users, how they browsed themselves through the application as well as where they mostly decided to exit the application. Usability could also be tested by identifying how many clicks are usually performed by the users to execute some functionality and also by analyzing how easily the users get to learn how to use the application.

This section has presented how we applied our suggested method and prototype for Usage Mining in the case of Exact Online. More specifically, we analyzed the usage of the trial version of Exact Online for the period of 4 months. We customized our prototype code in order to process the logs received from Exact Online and provided the stakeholders with the codes and documentation to perform the analysis in the future.

9. Discussion

In this paper we have investigated *how we can inspect software operation data, in order to gain knowledge about how the software is used by the end-users.* We reviewed related literature on software usage analysis. We constructed and presented a method that could be used to analyze how the end-users are using a software product. We explicated this knowledge by distinguishing four different categories (statistical summaries of sessions and users' behavior, factors that influence the customers' decisions, users profiles, and the most frequent navigation paths) and we presented the variables that need to be inspected through software operation data to later extract them. For this, we suggested three analysis tasks (classification analysis, users profiling, and clickstream analysis) and we proposed data mining techniques that can be used to perform each analysis task. The method and data mining techniques were evaluated through a prototype, which was developed in R, and was used in the case of the trial users of Exact Online. The classification analysis task yielded factors that influence the trial customers in their decision to convert, with 85.7% accuracy. In the users profiling task, five different clustering algorithms were tested to create profiles of the customers based on their navigation behavior. In the clickstream analysis, we built Markov Chains to create possible patterns that users follow when they browse the trial version of Exact Online, as well as to study the probability of using a specific feature, and to find the most common entry and exit points of the application.

The outcomes of this research enrich the domain of software usage analysis (El-Ramly & Stroulia, 2004; Simmons, 2006) and generally IS/IT use (Sun & Teng, 2012). Although data analysis techniques had been previously developed e.g. for software reengineering purposes (El-Ramly et al., 2009; Smit et al., 2008) they were hardly ever used to analyze the behavior of end-users while utilizing the software in the field (Kristjansson & Van der Schuur, 2009). Also, they did not provide any holistic approach to the various usage knowledge types (user profiles, most frequent navigation paths, etc.). On the other hand, the similar domain of web mining had met a lot of development in the web usage mining field (Cooley et al., 1997). Although a lot of lessons could be learned from this field, analysis of in-the-field usage for software products would have some significant differences. In this research, we revised data mining techniques, and we explored how we could employ them to extract usage knowledge from software operation data.

The Usage Mining Method and techniques that were suggested in our research can be used by the software vendors to inspect how end-users actually use the product, which functionalities they prefer, and what kind of paths they follow to perform a related task. They can also test how the end-users deal with the new features of an updated release and discover usability issues. The usage profiles can help reorganize the product functionalities in different packages as well as attract and retain customers more effectively. Finding what features the users prefer may also help determine future extensions.

While implementing this research, we were constrained by the availability of data mining functions in R for the implementation of the suggested techniques, by R's static memory allocation settings combined with the system resources that were available, and the limitation of data that we had available from Exact Online, due to privacy issues. The method and prototype were designed to analyze usage of Software-as-a-Service products, but they could easily be extended and adjusted for other types of software products.

Further research could be performed on the usage mining topic: First, the method and prototype should be further extended with analyzing operation data from other types of products. We would also like to test the techniques that were suggested but not implemented in the current prototype. Furthermore, the prototype could be evaluated by users in terms of usability. In addition, the usage analysis could be extended to incorporating also historical software operation information or other data (e.g. release schedules or bug tracker data). Finally, we would like to see how software vendors are utilizing the extracted usage knowledge.

References

Bowring, J., Orso, A., & Harrold, M. J. (2002). Monitoring deployed software using software tomography. *SIGSOFT Software Engineering Notes, 28*, 2–9.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth.

Brock, G., Pihur, V., Datta, S., & Datta, S. (2008). Clvalid: An r package for cluster validation. *Journal of Statistical Software, 25*(4), 1–22.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., et al. (2000). *CRISP-DM 1.0 Step-by-step data mining guide*. Technical report, The CRISP-DM Consortium.

Cooley, R., Mobasher, B., & Srivastava, J. (1997). *Web mining: Information and pattern discovery on the world wide web. Proceedings of the 9th IEEE international conference on tools with artificial intelligence (pp. 5–58)*. Los Alamitos, CA: IEEE Computer Society.

Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems, 1*(1), 5–32.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., & Spiegelhalter, D. J. (2007). *Probabilistic networks and expert systems: Exact computational methods for Bayesian networks* (1st ed.). Springer Publishing Company.

Dustdar, S., & Gombotz, R. (2007). Discovering web service workflows using web services interaction mining. *International Journal of Business Process Integration and Management, 1*, 256–266.

El-Ramly, M., & Stroulia, E. (2004). Mining software usage data. In *International Workshop on Mining Software Repositories in 26th International Conference on Software Engineering* (pp. 64–68).

El-Ramly, M., Stroulia, E., & Samir, H. (2009). Legacy systems interaction reengineering. In A. Sefah, J. Vanderdonckt, & M. C. Desmarais (Eds.), *Human-centered software engineering* (pp. 316–333). London: Springer.

Everitt, B., Landau, S., & Leese, M. (2001). *Cluster analysis*. An Arnold Publication.

Fayyad, U. M., & Irani, K. B. (1992). On the handling of continuous-valued attributes in decision tree generation. *Machine Learning, 8*, 87–102.

Field, A. (2009). *Discovering statistics using SPSS* (3rd ed.). SAGE Publications.

Germanakos, P., Tsianos, N., Lekkas, Z., Mourlas, C., & Samaras, G. (2008). Capturing essential intrinsic user behaviour values for the design of comprehensive web-based personalized environments. *Computers in Human Behavior, 24*(4), 1434–1451.

Giudici, P. (2003). *Applied data mining: Statistical methods for business and industry*. Wiley.

Grinstead, C. M., & Snell, L. J. (2006). *Introduction to probability*. American Mathematical Society.

Han, J. (2005). *Data mining: Concepts and techniques*. San Francisco, CA: Morgan Kaufmann Publishers.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *Hierarchical clustering*. New York: Springer.

Haykin, S. (1998). *Neural networks: A comprehensive foundation* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Hevner, A. R., March, S. T., Jinsoo, P., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75–105.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys, 31*(3), 264–323.

Jones, S., Cunningham, S., & McNab, R. (1998). An analysis of usage of a digital library. *Research and advanced technology for digital libraries* (Vol. 1513 of lecture notes in computer science, pp. 261–277). Berlin/Heidelberg: Springer.

Judd, T., & Kennedy, G. (2004). Basic sequence analysis techniques for use with audit trail data. *Journal of Educational Multimedia and Hypermedia, 17*(3), 285–306.

Junco, R. (2013). Comparing actual and self-reported measures of Facebook use. *Computers in Human Behavior, 29*(3), 626–631.

Kantardzic, M. (2002). *Data mining: Concepts, models, methods and algorithms*. New York, NY, USA: John Wiley & Sons, Inc.

Kaufman, L., & Rousseeuw, P. J. (2008). *Finding groups in data: An introduction to cluster analysis*. Hoboken, NJ: John Wiley & Sons.

Kohonen, T. (2001). *Self-organizing maps* (3rd ed.). Springer.

Kristjansson, B., & Van der Schuur, H. (2009). *A survey of tools for software operation knowledge acquisition. Technical report UU-CS-2009-028*. Department of Information and Computing Sciences, Utrecht University.

Lee, J., Podlaseck, M., Schonberg, E., & Hoch, R. (2001). Visualization and analysis of clickstream data of online stores for understanding web merchandising. *Data Mining and Knowledge Discovery, 5*(1), 59–84.

Lefngwell, D., & Widrig, D. (2003). *Managing software requirements: A use case approach* (2nd ed.). Pearson Education.

Lin, C., & Tsai, C. (2011). Applying social bookmarking to collective information searching (CIS): An analysis of behavioral pattern and peer interaction for co-exploring quality online resources. *Computers in Human Behavior, 23*(3), 1249–1257.

Liu, B. (2006). *Web data mining: Exploring hyperlinks, contents, and usage data (data-centric systems and applications)*. Secaucus, NJ: Springer-Verlag New York.

Maruster, L., & van Beest, N. (2009). Redesigning business processes: A methodology based on simulation and process mining techniques. *Knowledge and Information Systems, 21*(3), 267–297.

Meo, R., Lanzi, P., Matera, M., & Esposito, R. (2006). Integrating web conceptual modeling and web usage mining. In B. Mobasher, O. Nasraoui, B. Liu, & B. Masandm (Eds.), *Advances in web mining and web usage analysis* (pp. 135–148). Berlin/Heidelberg: Springer.

Nusayr, A., & Cook, J. (2009). AOP for the domain of runtime monitoring: breaking out of the code-based model. In *Proceedings of the 4th workshop on domain-specific aspect languages* (pp. 7–10). New York, NY: ACM.

Okazaki, S. (2007). Lessons learned from i-mode: What makes consumers click wireless banner ads? *Computers in Human Behavior, 23*(3), 1692–1719.

Park, S. C., & Ryoo, S. Y. (2013). An empirical investigation of end-users' switching toward cloud computing: A two factor theory perspective. *Computers in Human Behavior, 29*(2013), 160–170.

Petruch, K., Tamm, G., & Stantchev, V. (2012). Deriving in-depth knowledge from IT-performance data simulations. *International Journal of Knowledge Society Research (IJKSR), 3*(2), 13–29. http://dx.doi.org/10.4018/jksr.2012040102.

R Development Core Team (2008). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering, 14*(2), 131–164.

Ruvini, J. D., & Dony, C. (2001). *Learning users' habits to automate repetitive tasks (pp. 271–297)*. San Francisco, CA: Morgan Kaufmann Publishers Inc.

Sartipi, K., & Safyallah, H. (2009). Dynamic knowledge extraction from software systems using sequential pattern mining. *International Journal of Software Engineering and Knowledge Engineering, 20*(06), 761–782.

Shen, J., Fitzhenry, E., & Dietterich, T. G. (2009). Discovering frequent work procedures from resource connections. In *Proceedings of the 13th international conference on intelligent user interfaces* (pp. 277–286). New York: ACM.

Simmons, E. (2006). The usage model: Describing product usage during design and development. *IEEE Software, 23*(3), 34–41.

Smit, M., Stroulia, E., & Wong, K. (2008). *Use case redocumentation from gui event traces. Proceedings of the 12th European conference on software maintenance and reengineering (pp. 263–268)*. Washington, DC: IEEE Computer Society.

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P.-N. (2000). Web usage mining: Discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations Newsletter, 1*(2), 12–23.

Stieger, S., & Reips, U. D. (2010). What are participants doing while filling in an online questionnaire: A paradata collection tool and an empirical study. *Computers in Human Behavior, 26*(6), 1488–1495.

Sun, J., & Teng, J. T. (2012). Information systems use: Construct conceptualization and scale development. *Computers in Human Behavior*.

Tan, P. N., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining* (1st ed.). Boston, MA: Addison-Wesley Longman Publishing.

Therneau, T. M., & Atkinson, E. J. (1997). *An introduction to recursive partitioning using the RPART routines*. Technical Report 61, Section of Biostatistics, Mayo Clinic, Rochester. <http://www.mayo.edu/hsr/techrpt/61.pdf>.

Vaishnavi, V. K., & Kuechler, W. Jr., (2007). *Design science research methods and patterns: Innovating information and communication technology* (1st ed.). Boston, MA: Auerbach Publications.

Van de Weerd, I., & Brinkkemper, S. (2008). *Meta-modeling for situational analysis and design methods (pp. 38–58)*. Hershey: Idea Group Publishing.

Van der Aalst, W. M. P., & Weijters, A. J. M. M. (2004). Process mining: A research agenda. *Computers in Industry, 53*(3), 231–244.

Van der Schuur, H., Jansen, S., & Brinkkemper, S. (2010). *A reference framework for utilization of software operation knowledge. 36th EUROMICRO conference on software engineering and advanced applications (pp. 245–254)*. IEEE Computer Society.

Xie, X. L., & Beni, G. (1991). A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*(8), 841–847.

Xie, Y., & Phoha, V. V. (2001). *Web user clustering from access log using belief function. Proceedings of the 1st international conference on knowledge capture (pp. 202–208)*. New York: ACM.

Zaidman, A., Calders, T., Demeyer, S., & Paredaens, J. (2005). *Applying web mining techniques to execution traces to support the program comprehension process. Proceedings of the ninth European conference on software maintenance and reengineering (pp. 134–142)*. Washington, DC: IEEE Computer Society.