RUM++: A Log Mining Approach to Classify Users Based on Data Profile

34

Helton Franco de Sousa, Leandro Guarino de Vasconcelos, and Laercio A. Baldochi

34.1 Introduction

The pervasiveness of the Web has changed several aspects of modern life. As business, government and banking services became online, people have no other option but using the Internet. However, it takes more than an Internet connection and a hardware device to benefit from online services. Using a web application is far from trivial for a large amount of users, especially those who were not born in the digital age. On the other hand, those who were born after 1995 are usually very comfortable with the web. It is common that they help their parents and grandparents to perform tasks online [11].

As people grow older, motor control and visual acuity decreases. Moreover, cognitive abilities such as learning, memory retrieval and attention are impacted negatively. The first attempts to support users affected by age-related issues were based on heuristics, for example, "if the user presents visual acuity problems, make the UI elements bigger". The problem with this approach is that it causes trade-offs in the design, as changes in the application's interface may compromise other aspects of its design – navigation, for instance.

Another problem with this approach is that a "one size fits all solution" is not appropriate. Studies show that age, alone, has little or no impact on web literacy [5, 7]. Recent studies show that senior citizens that got old in contact with technology usually perform better than those who did not have this contact over the years [15]. Thus, low technological experience is also an issue that prevents users to perform well when using web applications.

A key point in the design of a web application is that it is, in general, targeted for a large audience of users. A study performed by Mbipom and Harper [14] shows that accessibility is correlated to good aesthetics only in web pages that present a clean design, with few interaction elements. However, most e-commerce applications present pages with tens of interaction elements [16]. Therefore, one cannot compromise the user experience of the majority of the users in order to make the application accessible.

An approach that seems promising to tackle this problem is the usage of multi-layered interfaces [12], which are designed to support users with different abilities, offering simplified interfaces designed for seniors and people with low web literacy, as well as full fledged interfaces designed for advanced users. A relevant issue in order to make the multi-layered interface approach feasible for web applications is transparently identifying the user, so as to provide the correct interface layer. To this end, it is paramount to track the user's actions in order to detect usage patterns.

In previous work, we proposed the RUM approach [19] – Real-time Usage Mining – which allows analyzing users' logs during navigation. Towards identifying behavior patterns, RUM exploits Web Usage Mining (WUM) techniques over client logs, i.e., logs collected on the user's browser. These fine-grained logs report details regarding each interaction, such as mouse and keyboard events. Moreover, it reveals which HTML element was the target of an event [1].

The RUM approach has proven to be effective in order to analyze the user behavior in web applications. However, an application specialist was required for defining usage

H. F. de Sousa
POSCOMP, Federal University of Itajubá, Itajubá, Brazil
e-mail: helton.franco@unifei.edu.br

L. G. de Vasconcelos
Brazilian Institute for Space Research, São José dos Campos, Brazil
e-mail: leandro.guarino@lit.inpe.br

L. A. Baldochi (✉)
Institute of Mathematics and Computing, Federal University of Itajubá, Itajubá, Brazil
e-mail: baldochi@unifei.edu.br

S. Latifi (ed.), *16th International Conference on Information Technology-New Generations (ITNG 2019)*,
Advances in Intelligent Systems and Computing 800,
https://doi.org/10.1007/978-3-030-14070-0_34

patterns and user profiles, and matching patterns to profiles. Therefore, our mining approach was application dependent.

This work reports an extension in our previous work called RUM++, which allows the detection of usage patterns associated to the elderly and to users that present low web literacy – the target users of our work – in any web application.

In order to achieve our goal, we reviewed the literature in order to understand the interaction problems presented by our target users. Then, we investigated which attributes of the client logs are related to these problems. Following, we performed an experiment to find out how these attributes vary among target users and regular users. Based on the results of this experiment, we defined usage patterns that characterize our target users. Finally, we tested RUM++ using real logs from a gamification application. Results showed that the found patterns are effective in order to identify our target users by analyzing their interactions in web applications.

This paper is organized as follow. Section 34.2 presents a literature review on aging and technology. Section 34.3 presents our previous work, which aims at understanding the behavior of users in order to support the construction of adaptive web applications. Section 34.4 presents an extension to our previous work that aims at supporting aging users. In Sect. 34.5 we present a case study in which RUM++ is used to classify 44 users of a gamification learning application. Following, in Sect. 34.6, we discuss the results of our study. Finally, Sect. 34.7 presents our concluding remarks.

34.2 Literature Review

The impact of aging in the usage of computer applications dates back to the 1980s [6,21]. This seminal research pointed out important findings regarding older users, such as they are more likely to commit mistakes [6], they take more time to complete tasks and to learn how to use new applications [21].

As expected, these problems increased with the advent of the Web [7, 17]. Experiments performed with old and young users using web applications reported differences regarding the usage of input devices, application browsing and to the amount of time needed to perform tasks.

Chaparro et al. [4] proposed an experiment where young and older adults had to perform click and click-drag tasks. The results showed that older users performed the tasks more slowly. The work by Carvalho et al. [2] reports that older users frequently lost the pointer of the mouse and spend time trying to find it, which negatively impacts their performance. When typing is concerned, older users present a low performance both in terms of speed and accuracy when compared to young adults [3].

More recently, researches have investigated the performance differences in touch interaction. Findlater et al. [9] showed that, when using touchscreen devices, older users were still slower than young users, however their performance were better than when using the mouse.

The research by Hwang et al. [10] suggests that low visual acuity and motor control negatively affects the performance of users. In their experiment, after magnifying the interaction targets, the time to complete a task has decreased 14% and the error rate dropped 50%.

The behavior of older users also present differences in web browsing. A study by Liao et al. [13] noted that, before an action, older users take into consideration all the possible options, thus taking more time, while younger users chose an interaction target more quickly. As a result, older users browse a small group of pages and present page re-visitation rate significantly smaller.

Finally, it is worth to notice that interaction issues related to age are aggravated when the user has low technological experience. Moreover, even younger users that present low technological experience may experience the difficulties faced by older users when using web applications [5, 7]. Therefore, both age and technological experience may be associated to low web literacy.

As a result of our literature review, we found out that older users and people with low technological experience present interaction issues associated to point and click [2,4], and that these issues may be associated to low visual acuity and motor control [10]. Moreover, issues associated to the usage of the keyboard [3] and to web browsing [13] was also reported.

34.3 The RUM Approach

In order to enhance the usability and the user experience in web applications, we proposed RUM [19], an approach that allows mining client logs in real-time with the aim of understanding the behavior of users and profiling them. As a result, application developers may profit from our approach to code adaptive web applications. Figure 34.1 depicts RUM's architecture, which is organized in five modules:

(1) **Log collection:** collects and stores the user's actions performed in the application's interface;
(2) **Task analysis:** provides remote and automatic usability evaluation during navigation;
(3) **Automated KDD:** detects behavior or usage patterns exploring the navigation history of past users. This module exploits KDD techniques to uncover patterns from logs.
(4) **Knowledge repository:** stores and processes the detected behavior patterns, using parameters provided by the application specialist;
(5) **Service:** listens to requests from the web application, providing information regarding the user's behavior during navigation.

In Fig. 34.1, the arrows depict the data flow among modules and the numbers on arrows indicate the flow sequence. Initially, as illustrated by arrow 1, the Logging module detects the user's actions on the application's interface, considering the specificities of the input device (desktop, tablet, smartphone). Following, as depicted by arrows 2 and 3, the detected actions are converted to logs in order to be processed by the Task Analysis and Automated KDD modules. The detected behavior patterns feed the Knowledge Repository module (arrow 4), which is responsible for defining the relevance of each pattern.

While the user is browsing, the web application may interact with the Service module to request information regarding the user's actions (arrow 8). Based on this information, preprogrammed interface adaptations may be triggered. Arrow 9 depicts the response for a given request. Possible responses are the last actions of the current user (arrow 5), the result of the usability evaluation during navigation (arrow 6), and the behavior patterns performed by the active user (arrow 7).

Therefore, RUM provides two main services to the application developer: (i) task analysis and (ii) usage patterns detection. The first one exploits previous work on usability evaluation [18]. The Task Analysis module evaluates the execution of tasks by calculating the similarity among the sequence of events produced by users and those previously defined by the application specialist. This service is relevant to detect users who are struggling to execute tasks.

The usage patterns detection, on the other hand, relies on a KDD process to uncover patterns from logs. The KDD process implemented in RUM finds sequential patterns, i.e., sequences of actions performed by users. The approach then relies on an application specialist to match sequential patterns to user profiles. Moreover, the specialist may associate a given action to be triggered when a user performs an action that is associated to her profile. It is important to notice that these actions are implemented by the application developer. Thus, the role of RUM is to inform the web application about the occurrence of an action.

The original KDD process implemented in RUM is not able to effectively detected the performance difference between two users that execute the same task, as its pattern detection mechanism only considers the difference in the sequence of actions performed by each user. In order to understand how well a user performs a task, it is paramount to analyze other information hidden in logs, such as the time to execute each task, the movement of the mouse, the amount of clicks, scrolling, and so far. In order to classify users according to performance profiles, we developed an extension to RUM's KDD module. The resulting approach was renamed RUM++.

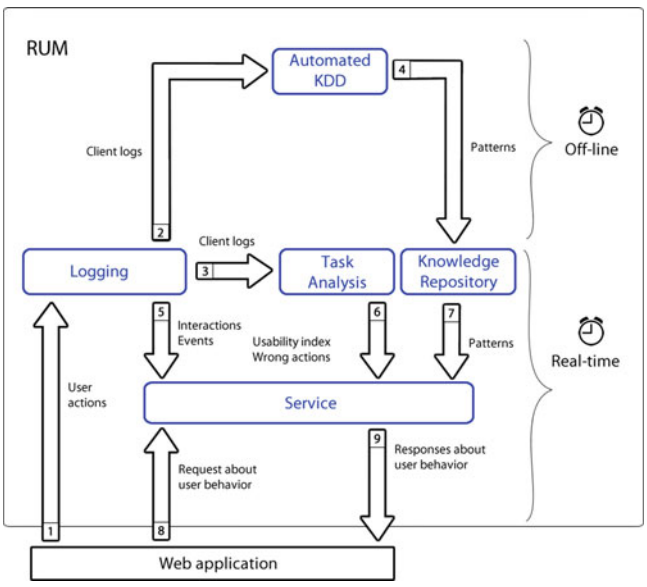

Fig. 34.1 Architecture of the RUM approach [19]

34.4 RUM++

In order to find patterns that may reveal the user ability when using a web application, we developed an approach based on the classical KDD process proposed by Fayyad et al. [8]. Our approach, depicted in Fig. 34.2, is composed of four steps, as follows:

(A) **Selection and Preprocessing:** Initially logs are cleaned up from irrelevant data and noise. Following, we extract attributes from logs and harness them to profile data collected from users.

(B) **Transformation:** The goal of this step is to reduce the dimensionality of the data to be analyzed. In order to

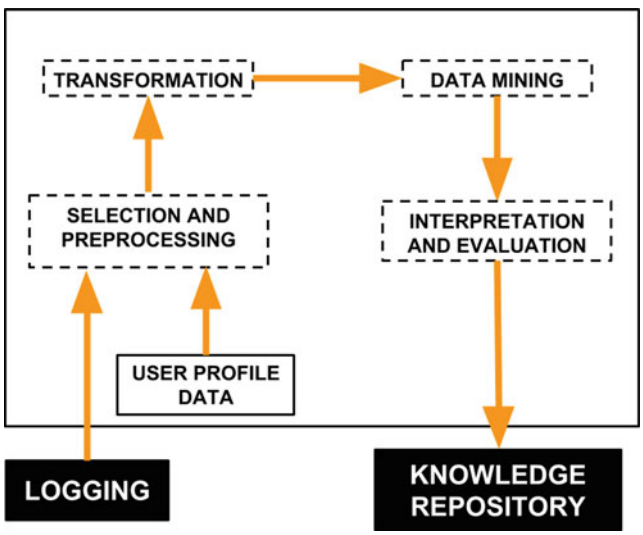

Fig. 34.2 Architecture of the RUM++ approach

achieve this goal, we select attributes that are correlated to the behavior of our target users.

(C) **Data Mining:** In this step, the selected attributes are used as input to data mining algorithms in order to classify users according to patterns presented in their logs.

(D) **Interpretation and Evaluation:** Finally, an specialist evaluates the mining results. In case the classification is correct, the used patterns are considered valid and, thus, are inserted in the knowledge repository.

As happens in the classical KDD process, our approach is iterative and incremental, possibly discovering new patterns as new data is processed. Following, we present the steps of our approach in detail.

34.4.1 Selection and Preprocessing

There are two input sources to our approach. The most important source are the collected logs, which present noisy data, such as requests to CSS files, javascript code and references to images. Thus, it is required to clean up logs from data that is not relevant for the mining process. The other data source is composed of user profiles, which need to be fed in the system in order to assure that the classification result is correct. Therefore, a task that needs to be done in this step is to harness logs to user profiles.

The log preprocessing is performed exploiting a tool for managing logs called Logstash. We use Logstash together with two other tools: a search engine called ElasticSearch and a tool for data visualization called Kibana. These tools form the so called ELK stack.[1] This way, after being preprocessed, collected logs are fed to Logstash, from where they can be searched using ElasticSearch. Finally, using Kibana one may visualize the logs and perform user friendly searches. We also provide a Python application in order to facilitate the execution of queries in the log data.

By the end of this step, the 20 following attributes are extracted from logs and made available.

(1) Amount of clicks;
(2) Amount of double clicks;
(3) Page elements associated to mouse interactions;
(4) Use of shortcut commands;
(5) Amount of zoom;
(6) Path within the application during navigation;
(7) Amount of key pressed;
(8) Sequence of events performed by the user;
(9) Amount of errors when filling forms;
(10) Orientation changes (for mobile devices);
(11) Mouse over amount for each page element;
(12) Total amount of events;
(13) Browsing duration;
(14) Amount of time spent per page;
(15) Scrolling speed;
(16) Amount of scroll;
(17) Typing speed;
(18) Amount of visited pages;
(19) Number of focus events;
(20) Number of visits per page.

[1] https://www.elastic.co/elk-stack

34.4.2 Transformation

In transformation, the goal is to reduce the dimensionality of our data. In order to do so, we need to select the log attributes that can reveal behavioral differences among our target users and regular users. In order to achieve this goal, we analyzed the work presented on Sect. 34.2 in order to understand which log attributes are related to the interaction problems discussed in the literature. As a result, a group of 12 attributes have been selected as candidates for mining. In order to make sure the relevance of these attributes, we used a correlation technique to ensure that our selection was correct.

As discussed in Sect. 34.2, older users and people with low technological experience present interaction issues associated to point and click [2, 4, 10], use of the keyboard [3] and browsing [13]. Moreover, the literature also report that interaction with small targets tend to be more difficult. Therefore, log attributes associated to these issues are important candidates for investigation. Based on these findings, we selected 12 attributes – *1, 2, 5, 7, 9, 12, 13, 15, 16, 17, 18 and 19* – from the total of 20 attributes generated in the preprocessing step.

The number of relevant attributes may be narrowed using the Pearson Correlation. The approach for this is to calculate the correlation coefficient among the 12 selected attributes and data features that represent our target users. Section 34.5 explains this procedure.

At the end of this step, log data is transformed into data which is ready to be used as input to the mining algorithms.

34.4.3 Data Mining

In order to facilitate our mining process, we resort to Weka, a machine learning workbench for data mining. Among the several state-of-the-art algorithms provided by Weka, we selected *RepTree* (Reduces Error Pruning Tree Classifier), as we needed to classify users according to patterns found in their logs. Moreover, *RepTree* is considered fast [20] and may be used with numeric data. This algorithm builds a

decision tree using information gain/variance and prunes this tree using reduced-error pruning.

From the available training options, we secleted *Cross-validation*. This technique allows evaluating the generalization ability of the model. The result of the classification provided by Weka is forwarded to the next step of our approach.

34.4.4 Interpretation and Evaluation

In the final step, the classification rules found by RepTree must be analyzed in order to verify if they are effective in order to classify users accordingly. In this step we need an application specialist to check if the classification provided by a given rule is effective. If so, the provided rule is added to the knowledge repository. Otherwise, it is discarded.

In order to evaluate RUM++, we performed an experiment with a real web application, involving 44 different users. The results of this experiment is discussed in the following section.

34.5 Case Study

In order to refine our approach, we performed an experiment using data from a real web application. As our aim was to evaluate how well RUM++ classifies users according to age and technological experience, we recruited users from two different group ages: the first group of users with age ranging from 18 to 39 years-old (group A), and second with ages ranging from 40 to 59 years-old (group B). In total, 44 subjects participated in the experiment, 16 from group A and 28 from group B.

Besides age, it was important to classify users according to their technological experience. In order to acquire this information, we developed a questionnaire in which users report the frequency (per week) that they use computers and smartphones, as well as the number of years using these devices. We also collected information regarding the level of education and the gender of each participant.

We used the number of years using computer devices, plus the frequency of use to compute a value between 0 and 1, which we called *Technological Experience Coefficient (TEC)*. A TEC value close to 0 means a subject with low technological experience and close to 1, a subject with high technological experience.

After classifying the recruited users according to their age and technological experience, we asked them to perform tasks in a learning gamification system called Level Learn (www.levellearn.com.br). The aim of this system is to support learning by providing an environment based on challenges and rewards. A challenge consists of a task to be performed in the system, and when the student performs the task correctly, she receives a reward.

In order to evaluate our users, we planned a challenge that required users to browse several pages, fill forms and interact with specific buttons in the user interface, sometimes requiring scrolling the page to find targets.

All 44 users performed the task using a desktop computer, with standard mouse and keyboard. Firstly, users were briefed about the Level Learn application, learning how to complete an example challenge. Following, the test challenge was explained and the user was left by herself to perform the required tasks. At the end, users were instructed to close the browser to quit the collection of logs.

34.6 Results and Discussion

All 44 participants succeeded in completing the task. Table 34.1 summarizes the data related to Young and Older users, in terms of mean age, amount of time using computer devices, frequency of use per week and the TEC calculated from this data.

As discussed in Sect. 34.4, we exploited the data on Table 34.1 to calculate the correlation among this data and the 12 attributes extracted from logs in the preprocessing step. Table 34.2 presents the results.

It is worth to notice that correlation values above 0.5 indicate significant positive correlation, while values bellow −0.5 indicate significant negative correlation. Interestingly,

Table 34.1 Collected data for young and older adults

	Young	Older
Age (years)	23.59	54.27
Usage time (years)	8.21	5.93
Usage frequency (per week)	6.62	2.4
TEC	0.88	0.44

Table 34.2 Correlation among log attributes and user data

	Description	Age	Frequency	Time	TEC
1	Amount of clicks	−0,213	0,199	0,100	0,166
2	Amount of double clicks	0,054	0,046	−0,165	−0,063
5	Amount of zoom	0,0	0,0	0,0	0,0
7	Amount of key pressed	0,597	−0,756	−0,689	−0,784
9	Amount of errors	0,178	−0,104	−0,046	−0,084
12	Total amount of events	0,569	−0,703	−0,654	−0,736
13	Browsing duration	0,753	−0,855	−0,641	−0,815
15	Scrolling speed *Scroll's*	−0,156	0,107	−0,066	0,026
16	Amount of scroll	0,648	−0,616	−0,601	−0,659
17	Typing speed	0,700	−0,750	−0,605	−0,737
18	Amount of visited pages	0,127	−0,101	0,011	−0,063
19	Number of focus event	−0,173	0,190	0,037	0,127

Table 34.3 Classification efficiency of RUM++ using REPTree

	Efficiency
Age	68.18%
Technological experience	84.09%

as shown in highlighted lines on Table 34.2, attributes 7, 12, 13, 16 and 17 correlates positively to age and negatively to usage time, usage frequency and technological experience. Hence, these attributes are more likely to reveal usage differences associated to young and older users. As a result, we were able to reduce the dimensionality of our data from 12 to 5 attributes, which makes the mining process much more efficient.

After concluding the transformation step, the resulting log data is used as input to the algorithm *REPTree* in Weka workbench using *Cross-Validation* with fold value 43 (dataset size - 1), with the aim of classifying users according to their age and technological experience. The efficiency of REPTree to classify users is shown in Table 34.3. As it can be noticed from the presented results, technological experience is more related to low web literacy than age. However, the experiment confirms that both of them impact the performance of users in web applications.

An interesting feature provided by Weka is the possibility to analyze the decision tree created by the REPTree algorithm. The tree created for classifying users according to age shows that the attribute 17 alone is enough to perform the classification. On the other hand, when the goal is to classify users according to their technological experience, two attributes are needed to obtain a good classification: 7 and 16.

Upon analyzing the classification results we concluded that the generated rules were sound and, therefore, we included them in the knowledge repository.

34.7 Conclusion

It is well known that the benefits brought about by web applications does not reach everyone. Aging users, for instance, usually struggle to perform tasks online. This work proposed an approach to support these users by providing a way to detect them transparently as they browse the web.

Our approach leverages the traditional KDD method proposed by Fayyad et al. [8]. Our main contribution lies in the preprocessing and transformation steps of the method, as we use both knowledge from the literature and a correlation technique to reduce the dimensionality of the data. The transformed data provided for the mining algorithm allows good results in the classification of users.

As future work, we plan to implement the analysis in real time, so as to support the construction of adaptive web applications.

References

1. Atterer, R., Wnuk, M., Schmidt, A.: Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In: Proceedings of the 15th International Conference on World Wide Web, WWW'06, New York, pp. 203–212. ACM (2006). ISBN:1-59593-323-9
2. Carvalho, D., Bessa, M., Magalhaes, L.: Different interaction paradigms for different user groups: an evaluation regarding content selection. In: Proceedings of the XV International Conference on Human Computer Interaction, New York, pp. 40:1–40:6. ACM (2014). ISBN:978-1-4503-2880-7
3. Carvalho, D., Bessa, M., Magalhães, L., Carrapatoso, E.: Age group differences in performance using diverse input modalities: insertion task evaluation. In: Proceedings of the XVII International Conference on Human Computer Interaction, Interacción'16, New York, pp. 12:1–12:8. ACM (2016). ISBN:978-1-4503-4119-6
4. Chaparro, A., Bohan, M., Fernandez, J., Choi, S.D., Kattel, B.: The impact of age on computer input device use:: psychophysical and physiological measures. Int. J. Ind. Ergon. **24** (5), 503–513 (1999)
5. Crabb, M., Hanson, V.L.: Age, technology usage, and cognitive characteristics in relation to perceived disorientation and reported website ease of use. In: Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility, ASSETS'14, New York, pp. 193–200. ACM (2014). ISBN:978-1-4503-2720-6
6. Czaja, S.J., Hammond, K., Blascovich, J.J., Swede, H.: Age related differences in learning to use a text-editing system. Behav. Inf. Technol. **8** (4), 309–319 (1989)
7. Fairweather, P.G.: How older and younger adults differ in their approach to problem solving on a complex website. In: Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility, Assets'08, New York, pp. 67–72. ACM (2008). ISBN:978-1-59593-976-0
8. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: The KDD process for extracting useful knowledge from volumes of data. Commun. ACM **39** (11), 27–34 (1996). ISSN:0001-0782
9. Findlater, L., Froehlich, J.E., Fattal, K., Wobbrock, J.O., Dastyar, T.: Age-related differences in performance with touchscreens compared to traditional mouse input. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'13, New York, pp. 343–346. ACM (2013). ISBN:978-1-4503-1899-0
10. Hwang, F., Hollinworth, N., Williams, N.: Effects of target expansion on selection performance in older computer users. ACM Trans. Access. Comput. **5** (1), 1:1–1:26 (2013). ISSN:1936-7228
11. Lara, S.M., Fortes, R.P., Russo, C.M., Freire, A.P.: A study on the acceptance of website interaction aids by older adults. Univers. Access Inf. Soc. **15** (3), 445–460 (2016). ISSN:1615-5289
12. Leung, R., Findlater, L., McGrenere, J., Graf, P., Yang, J.: Multi-layered interfaces to improve older adults initial learnability of mobile applications. ACM Trans. Access. Comput. **3** (1), 1:1–1:30 (2010). ISSN:1936-7228
13. Liao, C., Groff, L., Chaparro, A., Chaparro, B., Stumpfhauser, L.: A comparison of website usage between young adults and the elderly. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting. **44** (24), 4–101–4–101 (2000)

14. Mbipom, G., Harper, S.: The interplay between web aesthetics and accessibility. In: The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS'11, New York, pp. 147–154. ACM (2011). ISBN:978-1-4503-0920-2
15. O'brien, M.A., Rogers, W.A., Fisk, A.D.: Understanding age and technology experience differences in use of prior knowledge for everyday technology interactions. ACM Trans. Access. Comput. **4** (2), 9:1–9:27 (2012). ISSN:1936-7228
16. Paz, F., Paz, F.A., Pow-Sang, J.A.: Evaluation of usability heuristics for transactional web sites: a comparative study. In: Latifi, S. (ed.) Information Technology: New Generations, Cham, pp. 1063–1073. Springer International Publishing (2016). ISBN:978-3-319-32467-8
17. Priest, L., Nayak, L., Stuart-Hamilton, I.: Website task performance by older adults. Behav. Inf. Technol. **26** (3), 189–195 (2007) ISSN 0144-929X.
18. Vasconcelos, L.G., Baldochi, L.A., Jr.: Towards an automatic evaluation of web applications. In: SAC'12: Proceedings of the 27th Annual ACM Symposium on Applied Computing, New York, pp. 709–716. ACM (2012). ISBN:978-1-4503-0857-1
19. Vasconcelos, L.G., Baldochi, L.A., Santos, R.D.C.: Rum: an approach to support web applications adaptation during user browsing. In: Gervasi, O., Murgante, B., Misra, S., Stankova, E., Torre, C.M., Rocha, A.M.A., Taniar, D., Apduhan, B.O., Tarantino, E., Ryu, Y. (eds.) Computational Science and Its Applications – ICCSA 2018, Cham, pp. 76–91. Springer International Publishing (2018). ISBN:978-3-319-95165-2
20. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Series in Data Management Systems, 3rd edn. Morgan Kaufmann, Amsterdam (2011). ISBN:978-0-12-374856-0
21. Zandri, E., Charness, N.: Training older and younger adults to use software. Educ. Gerontol. Int. Q. **15** (6), 615–631 (1989)