

Tracking Mouse Movements for Monitoring Users' Interaction with Websites: Implementation and Applications

Urban Sedlar, Janez Bešter, Andrej Kos

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana, Slovenia

Abstract. The importance and ubiquity of the World Wide Web call for efficient design of websites and web-based applications, suitable both to inexperienced and possibly first-time users, as well as highly skilled users. However, assuring such design is not easy and is usually aided by some feedback mechanism, most commonly the monitoring of users' interaction with the site. With many methods in existence, there is a trade-off between accuracy and relevance on one hand versus the reach (the size of the population) and monitoring costs on the other. The most accurate method for monitoring user interaction is eye tracking, which provides detailed information on the distribution of user's attention, but is technically the most demanding. Based on earlier research, which provides evidence that mouse and eye movements on websites are highly correlated, we describe in this paper a method for tracking mouse movements on web sites. There are number of uses where the proposed method can provide relevant results. We describe three possible uses, demonstrating two of them by processing the data, collected on our website in a period of 200 days.

Key words: usability, accessibility, websites, mouse tracking, user modeling

Sledenje premikov miške za ugotavljanje interakcije uporabnikov s spletnimi stranmi: implementacija in uporaba

Razširjen povzetek. Pomembnost in široka navzočnost svetovnega spleta zahtevata učinkovito zasnovano spletnih strani in aplikacij, ki so primerne tako za neizkušene (nove) kot tudi za izkušene uporabnike. Kljub pomembnosti predstavitve na spletu je kakovost številnih strani in aplikacij slaba, v veliki meri zaradi neupoštevanja priporočil za uporabnost in dostopnost. Upoštevanje teh smernic postaja čedalje pomembnejše v času, ko se prebivalstvo stara, hkrati pa je čedalje več novih uporabnikov iz različno razvitih delov sveta. Za zagotavljanje uporabnosti in dostopnosti je potrebno spremljanje interakcije uporabnikov, bodisi z neposrednim opazovanjem bodisi prek posrednih tehnik. Med slednjimi je najpreprostejša analiza spletnih dnevniških datotek (datotek s podatki o vseh transakcijah spletnih strežnikov), najnatančnejša pa je metoda sledenja očem, ki z uporabo kamere beleži vsak premik oči in ga preračuna v koordinate na objektu, ki ga uporabnik opazuje. Prva tehnika je relativno nenatančna in ne omogoča spremljanja podrobnih interakcij, druga pa zahtevna za uporabo in se redko uporablja zunaj laboratorija, kar nujno pomeni testiranje uporabnikov v njim nepoznanem okolju. V članku predlagamo implementacijo in možnosti uporabe sistema za beleženje premikov miške na spletnih straneh. Jedro sistema je programska koda v jeziku JavaScript, ki se vstavi v telo spletne strani, ko do slednje dostopamo prek proxy strežnika (slika 1). Medtem ko uporabnik interagira s spletno stranjo, JavaScript program pošilja podatke o premikih miške, ki so opremljeni s časovnim žigom z natančnostjo 1 ms, na podatkovni strežnik. Podrobneje smo predstavili tri možnosti uporabe in dve od njih demonstrirali s procesiranjem podatkov, zbranih na spletni strani Laboratorija za telekomunikacije na Fakulteti za elektrotehniko. Zbrane po-

datke smo uporabili za prikaz zemljevida vročih točk na spletni strani, ki lahko neizkušenim uporabnikom pomaga osvetliti pomembnejše elemente na strani; z uporabo algoritmov za analizo podatkov o sledenju oči smo določili območja fiksacij kurzorja miške, predlagali pa smo tudi možnost uporabe podatkov za minimizacijo Fittsovega indeksa težavnosti, ki lahko olajša izvajanje pogostih opravil.

Ključne besede: uporabnost, dostopnost, splet, sledenje miške, modeliranje uporabnikov

1 Introduction

Presenting information on the World Wide Web is no longer just a matter of choice (for e-commerce and individuals alike), but in many cases a necessity that can draw the line between failure and success.

Yet the quality of presentation in many cases remains poor and does not follow usability and accessibility recommendations. This is becoming increasingly important as the structure of internet users changes and the proportion of older population grows. It is estimated that the number of internet users will double in the next ten years, the majority coming from Asian countries; that may bring about significant changes in worldwide usability needs [1].

The need for adequate usability and accessibility has become even more pronounced with the emergence of highly interactive web applications enabled by asynchronous means of communication with the server (i.e. AJAX - Asynchronous JavaScript and XML), dubbed “Web 2.0”. Functionalities and performance of such web-based applications are now more or less comparable to ordinary client applications, the only difference being the use of regular web pages as a user interface.

An important method for usability evaluation of websites and web applications is analyzing users’ interaction with the site. This can be performed in a variety of ways, ranging from simple web server log analysis to highly accurate eye tracking.

However, simple methods (such as log analysis) do not provide adequate accuracy to get meaningful results and the highly accurate methods (such as eye tracking) require considerable resources and are usually performed in a laboratory, where users might act differently than at the workplace or in the privacy of their homes.

In this paper we discuss a method for accurate and unobtrusive monitoring of the users’ activity on websites by tracking their mouse movements, as well as some of its possible uses, such as identifying the level of users’ experience or presenting an aid for inexperienced users.

2 Problem statement

Web site usability evaluation in general largely relies on observations of real users’ interaction. Preferably, users’ gazes are traced, as they reveal the exact positions that attract their attention (i.e., eye fixations).

It is very important that observations are performed in the users’ natural environment (i.e. at work, home), where results tend to be more relevant. Data gathering and analysis should be as unobtrusive as possible, requiring no special efforts at the users’ side.

The cost of usability testing should be low and should preferably not increase linearly with the number of users monitored.

3 Existing solutions

Measuring the users’ interaction with websites is currently possible by using different methods, such as web server log analysis, client-side scripting-based tools, eye tracking and mouse tracking, each being described briefly in this section.

3.1 Web server log analysis

A typical web server allows logging of all transactions (requests from web browsers) to a file. As this feature is commonly enabled, information about the transactions is readily available or can be made so relatively easily.

Information in the log can include any of the following:

- date and time of the transaction,
- the file requested,
- requesting IP address,
- user agent (browser) information, and
- referring page address.

This information can be used for statistical purposes, such as counting the number of requests and the number of visits (each visit consists of one or more requests), or to infer geographical locations of users (based on IP geo-location), page popularity and search engine queries that drive the traffic to the site (acquired from referring page addresses).

This method records information without users’ knowledge or explicit permission and has great reach, but although simple to deploy and extract information from, it cannot offer more fine-grained information, such as which part of page or application drives traffic or causes problems.

3.2 Client-side scripting-based analysis tools

Wide availability of client-side scripting has paved the road for more detailed tracking and many services available free of charge exist today (e.g. Google analytics).

The underlying philosophy of this approach is injecting a small piece of the scripting code (usually JavaScript) into the body of a web page; when the page loads, the program code is executed, posting some data, available to the browser, back to the server. The server collects the data and displays it on the administrator’s request.

This enables gathering of more detailed data, ranging from screen resolution and operating system, to specific browsing patterns (e.g. returning visits, based on cookie identification).

However, no publicly available system tracks users’ interaction with the content itself; furthermore, the tracking code has to be injected manually into every page by the site administrator, thus administrative rights are needed and only monitoring of one’s own sites is possible. This prevents usability studies of third party sites.

3.3 Eye tracking

Eye tracking is a technique commonly used in cognitive studies, human-computer interaction research and marketing research. It is performed using an eye tracker, a device that usually consists of an infra-red camera which focuses on one or both eyes and records the movements as the viewer is looking at the stimulus.

An eye tracker can be head mounted; if not, it requires the head to be stabilized. In addition, it needs to be calibrated prior to monitoring to provide accurate results.

Recorded eye movements can be analyzed and divided into fixations, the areas where the eye fixates (pauses), and saccades, areas where the gaze drifts to another position. The eye tracker records eye position with adequate frequency (50 Hz or more) to capture very rapid eye movements, thus providing a detailed map of users' attention.

The downside of using an eye tracker for monitoring users is complicated setup and need for constant recalibration [3]. As a consequence, only a limited number of users can be monitored as the cost is high and it increases linearly with the size of the population.

Another disadvantage is the testing environment in which users are rarely as comfortable as at work or in the privacy of their homes. In addition, the mere awareness of the fact that one is being monitored can change his/her behavior significantly.

3.4 Mouse tracking

Recently there have been attempts to implement a mouse tracking application using client-side scripting (see [2], [4], [5]). However, none of them provides high-accuracy time-stamped data with temporal resolution comparable to eye trackers. Furthermore, some solutions ([2] and [5]) rely on client-side scripting to process the data and post only "mouse fixations" to the server, which renders a detailed analysis impossible.

4 Proposed solution

As previous work [3] reports a considerable correlation between eye movements and mouse movements, it becomes reasonable to consider tracking mouse movements as an option that could supplement eye tracking.

The proposed solution tracks mouse movements, left and right clicks and page scrolling on any web page, thus including also third-party sites. Temporal resolution for such tracking can be as high as 1 ms.

The main component of the proposed system is a client-side browser script (written in JavaScript), which is embedded into the webpage and is started when the page loads. Since the majority of contemporary web applications rely on heavy use of JavaScript, there is little concern that JavaScript interpretation would be disabled in an average browser.

The advantage of such implementation is the ability to simultaneously collect data from an arbitrary number of users at virtually no cost. The users are at all times in their everyday environment and can even not be aware that their every move is recorded. Although the relevance

of thus collected data is high, the privacy concerns have to be taken into consideration.

5 Implementation

The system in whole comprises the following three parts:

- JavaScript program code running in the web browser collects the mouse activity data and sends it to the database server,
- database server receiving the data stores it in the database, and
- proxy server facilitating the use of the system by inserting the JavaScript code into any page accessed through it.

The system architecture and the process of tracking the mouse activity are presented in Figure 1.

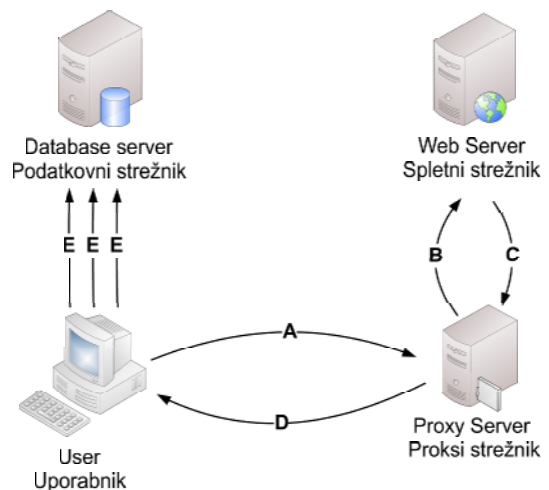


Figure 1. **System architecture and the process of mouse activity tracking** (A – user requests a web page via proxy; B – proxy requests the page from web server; C – web server returns the page; D – proxy inserts tracking JavaScript code and returns the page to user; E – JavaScript code runs inside the browser and sends the information about mouse movements to database server).

Slika 1: **Arhitektura sistema in proces sledenja premikov miške** (A – uporabnik preko proxy strežnika zahteva spletno stran; B – proxy strežnik zahteva stran od spletnega strežnika; C – spletni strežnik vrne zahtevano stran; D – proxy strežnik vstavi JavaScript kodo za sledenje premikov miške in vrne stran uporabniku. E – JavaScript koda teče v spletnem brskalniku in pošilja podatke o premikih miške na podatkovni strežnik).

5.1 JavaScript program code

JavaScript is a client-side scripting programming language which has become a fairly standard part of most modern web browsers. The code is embedded in or included from an HTML page and allows basic manipulation of the document, as well as fetching and posting

information from and to the server or capturing user interface events.

When an event (*MouseMove*, *MouseDown* or *MouseUp*) occurs, the capture function is evoked, saving the coordinates and mouse button status to an array, along with precise timestamp in Unix time (number of milliseconds since midnight, Jan. 1, 1970). When the size of the array exceeds $N = 50$ coordinates, the data is posted to the server.

Other triggers for posting are possible (and were used before [2], [5]), for example clicking a link on the page; unfortunately this fails when the user leaves the page without clicking a link (e.g., by clicking the browser “back” button).

The data, collected and sent to the server, additionally includes user-agent information (browser version, screen resolution, operating system) and cookie identification, which - along with the client IP - enables association of collected information with prior sessions for returning users.

5.2 Proxy Server and Database Server

The JavaScript code of the implemented mouse tracking system is included from the accessed (i.e., monitored) home page using an *include* statement of the following form.

```
<script type='`text/javascript`'  
  src='`http://server.com/tracker.js`'  
</script>
```

The statement is added to the home page by the proxy server, that the page is accessed through, along with a reference to an empty and hidden image, similar to the following.

```
<img id='`foo`' name='`bar`'  
  style='`display:none`' src='`' />
```

The image tag is used for the purpose of posting the data, as it remains the only way of circumventing JavaScript security constraints which prevent the script to post the data to a non-originating server. Using the image tag, the source of the picture can be changed to an URL which bears the information about the mouse position, for example:

```
http://server.com/save.php?x=100&y=100
```

The server-side script on the database server that receives the query, carrying information about the mouse position, returns an empty image of the dimension of 1-by-1 pixels and saves the mouse position into the database.

6 Use of the collected data

The information gathered can help website developers to identify the major usability and accessibility flaws, but has many other applications as well:

- identifying the level of the users’ experience with the site and displaying usage tips or presenting them with a basic design instead of an interface for advanced users,
- displaying mouse-movement activity level of an average user to aid inexperienced users,
- detection of pen-based interaction from lack of continuous traces and providing an interface better suited to the use of pen,
- learning the patterns of user interaction and using them to aid authentication (see [6]) along with cookies, IP address and user credentials,
- gathering precise usage data as a measure of the users’ level of interest in a specific content (incl. advertisements) by identifying mouse fixations,
- dynamically (or manually) adapting user interface based on the user’s interaction pattern (e.g. rearranging the elements of the webpage to minimize Fitts’ *index of difficulty* to perform an average task), and
- *caching* of web page elements or linked pages based on prediction of mouse movements.

We focused on three sample cases: providing some information about average user’s behavior for inexperienced users, identifying mouse fixations and proposing minimization of the Fitts’ Index of Difficulty.

6.1 Case I: Aid for inexperienced users

We employed the proposed solution to gather mouse movement data on the website of Laboratory for Telecommunications, Faculty of Electrical Engineering at the University of Ljubljana (Fig. 2). The process lasted for a total period of 200 days (from Dec. 20, 2005 to Jul. 8, 2006). In that time, actions of over 5000 users, generating approximately 25.000 sessions, were tracked with temporal resolution of 1 ms. During the visits, users were not aware that their mouse movements were being tracked – such monitoring is possible only with the consent of the site owner. The ethical dimension of such monitoring was taken into consideration and caused some hesitation. However, the process was justified by an analogy of a shop owner, observing paths of anonymous customers to find out where the most desired items are located.

The data was used to generate a heatmap of the movement activity based on the number of mouse passes over

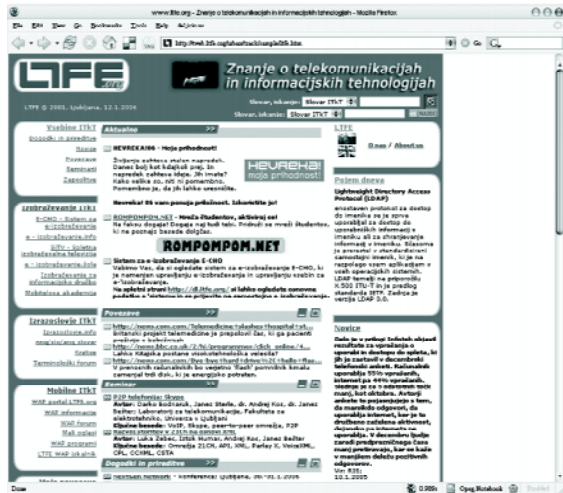


Figure 2. Home page of Laboratory for Telecommunications, used for monitoring of users' interactions.

Slika 2: Stran Laboratorija za telekomunikacije, uporabljena za spremljanje interakcij uporabnikov.

each pixel (Fig. 3). A higher user activity implies a potentially high level of interest of an average user; such heatmap, overlaid over the page, would aid inexperienced first-time users and could appear automatically after low level of experience was determined using techniques of pattern recognition.

6.2 Case II: Detecting mouse fixations

The information from a single user (over all of her/his sessions) can be extracted using a cookie-based identification. Algorithms, equivalent to those for fixation detection in eye tracking (see [8]) can be employed to determine areas of the page where the mouse fixates.

We employed the simplest technique, i.e. a velocity-based approach, determining the fixation as an area, where point-to-point velocity drops under a certain predefined threshold (Fig. 4). Velocity was determined by dividing the distance between consecutive coordinate pairs by the difference in timestamps.

The fixations determined in the process can be used in a similar manner as the eye fixations, which reflect the distribution of user's attention, but are obtained at virtually no cost and simultaneously for any number of users.

6.3 Case III: Minimizing Fitts' Index of Difficulty

The information about mouse clicks can be used to facilitate the use of the interface for an average user.

By using a model for the act of pointing (applicable to pointing with a computer mouse) called the Fitts' Law, we can calculate an index of difficulty (*ID*) to complete

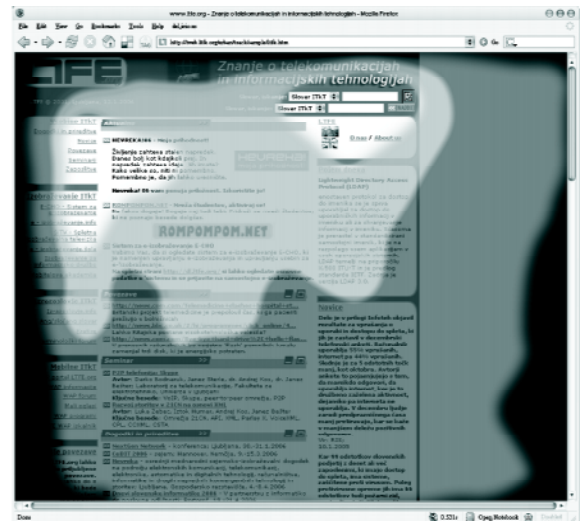


Figure 3. Homepage, overlaid with heatmap of high activity areas, based on the activity of all users over the period of 200 days. The image was generated by taking the frequencies of users' mouse passes over each pixel and dividing them into four groups, sized in geometric proportion, each represented with a different color. High-frequency noise was removed and pixel values were further linearly quantized into four categories, each assigned a different transparency level.

Slika 3: Stran, prekrita z zemljevidom vročih področij, ugotovljenih na podlagi aktivnosti vseh uporabnikov v obdobju 200 dni. Slika je bila ustvarjena na osnovi števila prehodov miške preko vsakega piksla; pogostost prehodov smo razdelili v štiri skupine v geometrijskem razmerju in vsako predstavili s svojo barvo; po odstranitvi visokofrekvenčnega šuma smo vrednosti pikselov linearno kvantizirali v štiri skupine z različno prosojnostjo.

a movement along a single dimension,

$$ID = \log_2 \left(\frac{2D}{W} \right)$$

where *D* represents the distance between the starting point of the movement and the center of the target, and *W* is the width of the target measured along the axis of motion (Fig. 5). *ID* can be used to estimate the time needed to complete the movement. The Fitts' law was extended to model two-dimensional tasks [7].

By finding patterns in user interaction, it is possible to identify the tasks having unnecessarily high *ID* and modify the interface to decrease it and make the page easier to use.

7 Conclusions

In the paper we present implementation and possible use of a system for monitoring user interactions on websites by employing a technique of tracking mouse movements. The system provides more accuracy than most of the commonly used and widely deployed analytical tools and could present a supplement to eye tracking. The data can be obtained without modifying the monitored site if the

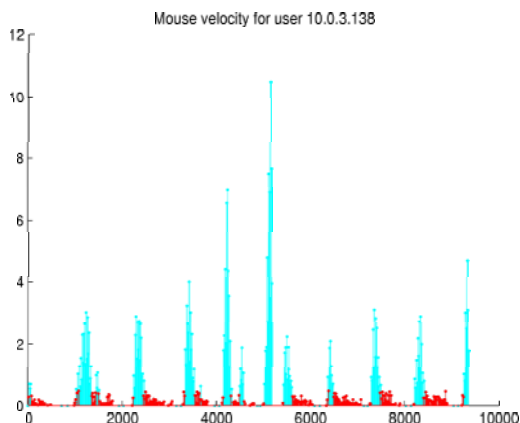


Figure 4. Graph of mouse velocity (in pixels/ms) versus time (in ms). Mouse fixations determined by using the velocity-based approach (with a threshold of 0.5 pixels per millisecond) are graphed in black.

Slika 4: Graf hitrosti miške (v piksljih/ms) v odvisnosti od časa (v ms). Fiksacije miške, določene z metodo na osnovi hitrosti (ko se hitrost miške spusti pod prag 0.5 piksla na ms), so prikazane v črni barvi.

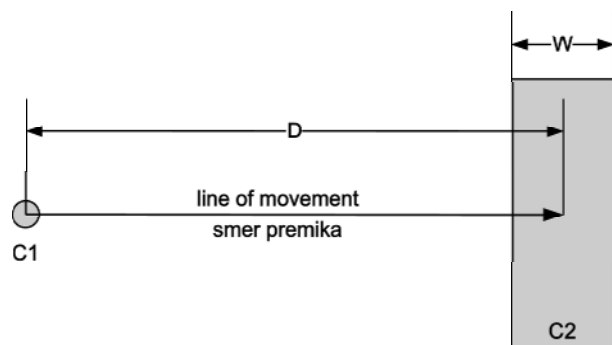


Figure 5. Meaning of symbols for evaluation of the Fitts' Index of difficulty (ID) for a one-dimensional task; suppose the website requires the user to make a click anywhere in the rectangle $C2$ when the mouse pointer resides at the point $C1$ (where the previous click was performed). The ID increases with the distance D and decreases with the width of the target W .

Slika 5: Pomen oznak v enačbi za oceno Fittsovega indeksa težavnosti (ID) za nalogo v eni dimenziji; predpostavimo, da spletna stran od uporabnika zahteva klik kjerkoli v pravokotniku $C2$, kurzor miške pa se nahaja na točki $C1$, kjer se je izvršil predhodni klik. Indeks težavnosti se povečuje z razdaljo D in zmanjšuje s širino tarče W .

user uses chooses to use a proxy or by modifying the site itself by an administrator. The collected data can be used in a variety of scenarios.

8 References

[1] Nielsen, J. *Alertbox, Dec. 19, 2005: One Billion Internet Users*
http://www.useit.com/alertbox/internet_growth.html, accessed Jul. 31, 2006

- [2] Mueller, F. and Lockerd, A. 2001. *Cheese: tracking mouse movement activity on websites, a tool for user modeling*. In CHI '01 Extended Abstracts on Human Factors in Computing Systems (Seattle, Washington, March 31 - April 05, 2001)
- [3] Chen, M. C., Anderson, J. R., and Sohn, M. H. 2001. *What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing*. In CHI '01 Extended Abstracts on Human Factors in Computing Systems (Seattle, Washington, March 31 - April 05, 2001)
- [4] Atterer, R., Wnuk, M., and Schmidt, A. 2006. *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 (Edinburgh, Scotland, May 23 - 26, 2006).
- [5] Arroyo, E., Selker, T., and Wei, W. 2006. *Usability tool for analysis of web designs using mouse tracks*. In CHI '06 Extended Abstracts on Human Factors in Computing Systems (Montral, Qubec, Canada, April 22 - 27, 2006).
- [6] Pusara, M. and Brodley, C. E. 2004. *User re-authentication via mouse movements*. In Proceedings of the 2004 ACM Workshop on Visualization and Data Mining For Computer Security (Washington DC, USA, October 29 - 29, 2004).
- [7] MacKenzie, I. S. and Buxton, W. 1992. *Extending Fitts' law to two-dimensional tasks*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Monterey, California, United States, May 03 - 07, 1992).
- [8] Salvucci, D. D. and Goldberg, J. H. 2000. *Identifying fixations and saccades in eye-tracking protocols*. In Proceedings of the 2000 Symposium on Eye Tracking Research & Applications (Palm Beach Gardens, Florida, United States, November 06 - 08, 2000).

Urban Sedlar graduated from the Faculty of Electrical Engineering at the University of Ljubljana, Slovenia, in 2004. He currently works towards his PhD as a junior researcher in the Laboratory for Telecommunications at the same faculty. His research focuses on web technologies, web-based services and multimedia. He is a student member of IEEE and ACM.

Janez Bešter received his Ph.D degree in the field of telecommunications from the Faculty of Electrical Engineering at the University of Ljubljana, Slovenia. He is currently the Head of the Laboratory for Telecommunications and an Associate Professor at the same faculty. His work focuses on planning, realization and management of telecommunication systems and services as well as application of information technologies to education. He is a member of AAATE, IEEE, IFIP, ACM and IEICE.

Andrej Kos graduated from the Faculty of Electrical Engineering at the University of Ljubljana, Slovenia in 1996. In 2003 he received his Ph. D. degree in telecommunications. He works as an Assistant Professor at the same faculty. He has conducted extensive research and has gained industrial experience in the analysis, modeling and design of advanced telecommunication elements, networks and systems. His current work and research focuses on next-generation packet switching and management. His other fields of expertise include mobile communications and intelligent networks. He is a member of IEEE, IEICE and the Telemanagement Forum.