функціями;

- хороша графічна візуалізація подання даних та результатів їхнього аналізу;
- можливість самостійного написання необхідних функцій;

Серед недоліків необхідно відзначити те, що на відміну від більшості комерційних програм, R має не графічний інтерфейс, а інтерфейс командного рядка, тому треба знати необхідні для роботи функції та синтаксис мови програмування (цей недолік дещо нівелюється встановленням пакету Rcmdr).

Дані, окрім введення з клавіатури, можна імпортувати з txt- та xml-файлів, Веб-сторінок, MS Excel, SPSS, SAS,Stata, систем керування базами даних, включно з Microsoft SQL Server, Microsoft Access, MySQL, Oracle, PostgreSQL. Також можна дані експортувати з R, зокрема в текстовий файл з розділювачами, таблицю Excel або файл іншого статистичного пакету (SPSS, SAS, Stata).

Зазвичай будь-яке дослідження завершується публікацією результатів. Для інтеграції програмного коду R та результатів дослідження в LaTeX використовується пакет sweave, а для експорту в документи формату ODF (Open Documents Format) – пакет odfWeave.

Джерела:

1. Роберт И. Кабаков R в действии. Анализ и визуализация данных в программе R / пер. с англ. Полины А. Волковой. – М.: ДМК Пресс, 2014. – 588 с.

Software system for parallel usability testing

Kostiuk D.A., Latiy O.O., Markina A.A.

Brest State Technical University, dmitriykostiuk@gmail.com

An open source software suite is presented targeted at evaluation of the effectiveness of human-computer interaction by monitoring physical state of the user. An internal architecture and capabilities of the system are reviewed.

Current approaches for evaluation of software usability (as ones based on cognitive schemes analysis, so ones based on expensive measuring tools like eye capture and medical-grade electroencephalography) can't be used without an expert's judgment after examination of collected data [1]. This way of evaluation is slow, prone to human errors and even to biased view in some cases. However, last years have brought into wide usage some computer-connected measuring devices, like pulsometers and simple electroencephalographs. Being targeted at sports and entertainment, these devices at the same time have enough precision to provide data of user's physical state during working with software. This provides the possibility to create totally new style of usability evaluations: fully automatic numerical estimation based common tests set and data acquired from popular measuring devices.

While carrying out some research in usability of GNU/Linux GUI, we have created specialized package of testing instruments and have used it in several experiments, which demonstrated proposed approach to be promising. These experiments have resulted in an integrated software suite for such tasks as running tests, acquiring measured data and accumulating them in central database, and what is more, providing parallel testing of several users to speed up experiments. It was implemented as an open source project titled "UXDump Suite" and is available at https://bitbucket.org/AsyaAliset/uxdump.

The architecture of the system is presented at figure 1. The main part of the system is written in C++ and Qt, and currently is compiled for two platforms, GNU/Linux and MS Windows, to compare usability of these platforms software.

Being targeted at parallel usage at several computers, it relies on MySQL database to store measurement results, which is supposedly residing on a separate server. Each computer participating in research has a copy of UXDump Suite, executed to carry on test with chosen measuring modules (for user-wearable sensors) chosen testing module and filled-up user's data (figures 2-b and 2-c). Same application is used to visualize portion of data (figure 2-a) with one of typical plot styles. Chosen data can be exported either as graphics (SVG and bunch of raster formats are supported) or as CSV table.

The architecture of the system is presented at figure 1. The main part of the system is written in C++ and Qt, and currently is compiled for two platforms, GNU/Linux and MS Windows, to compare usability of these platforms software.

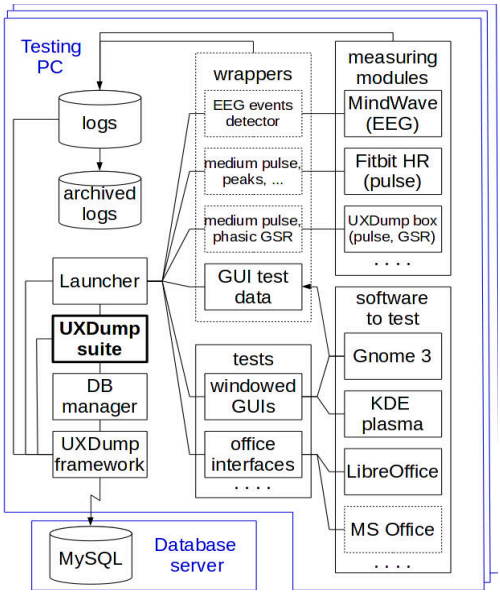
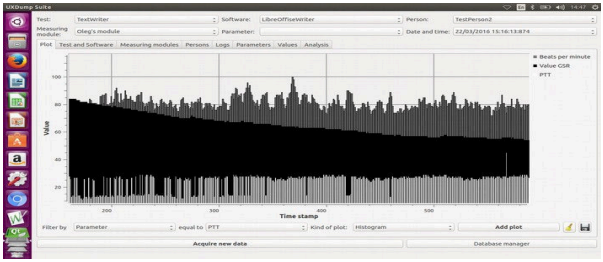


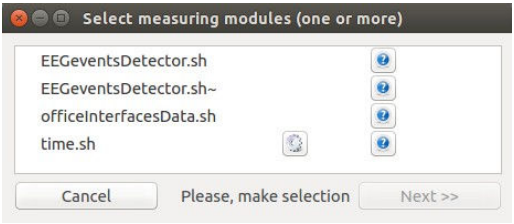
Figure 1 — Architecture of the system

Being targeted at parallel usage at several computers, it relies on MySQL database to store measurement results, which is supposedly residing on a separate server. Each computer participating in research has a copy of UXDump Suite, executed to carry on test with chosen measuring modules (for user-wearable sensors) chosen testing module and filled-up user's data (figures 2-b and 2-c). Same application is used to visualize portion of data (figure 2-a) with one of typical plot styles. Chosen data can be exported either as graphics (SVG and bunch of raster formats are supported) or as CSV table.

UXDump relies on a specific structure of subfolders.

Testing module checks whether the software to test is running (optionally starts it), and supplies the user with tasks to do. In fact these “modules” are subfolders of the `Test` folder, and contain at least one test starting script (a script with “start_” prefix). Currently two groups of tests are implemented. `WindowedGUIs` group includes 3 test variants to evaluate effectiveness of windows management in GUI shells [2], and `OfficeInterfaces` group contains 8 variants to evaluate usability of office applications [4]. Tests from the first group depend on Gnome 3, KDE Plasma Workspace, and Ubuntu Unity, while tests from the second group rely on LibreOffice (Writer and Calc), Calligra Suite (Words and Sheets), Microsoft Office 2007+ (Word and Excel), and two graphical editors (Pinta and Microsoft Paint). Of course, Microsoft-depending tests are available only if UXDump Suite is compiled for Windows. Figure 2-c shows starting dialog of the chosen testing module with specified user login and software (in combobox filled up by scanning starting script names).





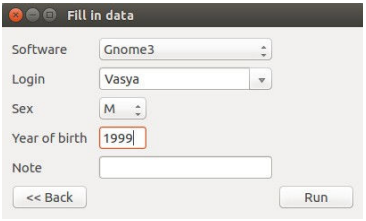


Figure 2. UXDump Suite: starting screen with plot (a), and test launcher asking to choose measuring modules (b) and to fill in user data before running test (c).

Measuring modules accept user's physical state data from wearable devices. In reality they are either executable files or scripts placed in the `Modules` folder. Moreover, some of them are not more than wrappers for real measuring modules, residing in deeper level subfolders. Such a cascade approach allows to separate acquiring raw data and its primary processing. For example, not yet written EEG MindWave module wrapper can be used to detect additional events (eye blinks), while `phasicGSR` wrapper is useful to extract phasic component of the skin galvanic response (GSR).

Currently following modules are designed to be used by the system: `fitbit` to measure heart rate (HR) with Fitbit Charge HR fitness tracker (based on open source galileo project), `mindwave` (based on the same-name open source library) to evaluate mind concentration with mass-market Neruosky Mindwave EEG headset, and `uxdumpbox` to acquire HR and GSR with our own open hardware project, the UXDump Box [3]. There are also two quasi-modules, `time` to register the experiment duration, and `gui_test_data` to get additional logs from the WindowedGUIs testing module.

Data are provided by measuring modules as log files in CSV format. File name is specified by the launcher component, which controls the process of testing: asks user to choose measuring modules and one testing module, runs all chosen modules, waits until testing module finishes its work and forces quit of measuring module processes. After data acquisition log files are stored in the local archive for further analysis if needed. CSV parsing and dealing with database is done through the set of libraries, specified on the figure 1 as UXDump framework. The same libraries are used by the DB Manager service application, targeted at administration of measurements stored in a remote database.

References

1. Paas F., Renkel A., Sweller J. Cognitive Load Theory: Instructional Implications of the Interaction between Information Structures and Cognitive Architecture // Instructional Science. No. 1-2(32), 2004. – pp. 1–8.

2. Kostiuk D.A., Derechennik S.S., Shitikov A.V., Latiy O.O. Approach to evaluate effectiveness of human-computer interaction with contemporary GUI // Третя міжнародна науково-практична конференція FOSS Lviv 2013: Збірник науковых праць, Львів, 18–21 квітня 2013 р. – Львів, 2013. – С. 85–87.

3. Костюк Д.А., Латий О.О. Оценка состояния пользователя с помощью платформы Arduino // Информационные технологии и системы 2014 (ИТС 2014): материалы МНК. Минск, БГУИР, 29 октября 2014 г. – С. 57–58.

4. Костюк Д. А., Латий О. О., Маркина А. А. Инструментальная оценка состояния пользователя в задаче сравнения интерфейсов офисных приложений // XII конференция разработчиков свободных программ. Тезисы докладов. – Калуга, 16-18 октября 2015г. – М.: Альт Линукс, 2015. – С.8–12.