

Testing Computer Skills In Productive Environment: Solution And Know-How

Andris Paikens, Zane Bicevska, Janis Bicevskis

Datorikas Instituts DIVI, Avotu 40-34, Riga, Latvia

Andris.Paikens@di.lv

The research discusses issues of training the computer skills and testing them in productive environment. Based on outcomes of the research project, authors describe recommendable solution for mastering office software applications skills and aspects of putting this solution into practice. Main conclusions of the research are based on data acquired and analyzed during approbation of solution in practice.

Keywords: digital literacy, testing of IT skills, eLearning.

1 Introduction

The eLearning initiative of the European Commission seeks to mobilize the educational and cultural communities, as well as the economic and social players in Europe, in order to speed up changes in the education and training systems for Europe's move to a knowledge-based society [1].

Modern, effective education and training systems are vital to everything from economic competitiveness to social inclusion [2]. Information and communications technologies (ICTs) are part of the answer, improving classical education and providing flexible learning solutions to people throughout their life. Vivianne Reding, Member of the European Commission responsible for Information Society and Media, announced: "ICTs can help improve education, life-long learning and social inclusion" [3].

Widespread digital literacy, however, is vital. To participate and take advantage, citizens must be digitally literate – equipped with the skills to benefit from and participate in the Information Society. Today, almost all workers need to be able to use ICTs, so training in using ICTs is both a key part of Europe's inclusion strategy and essential to reaping the benefits ICTs bring to education. The digital literacy includes both the ability to use new ICT tools and the media literacy skills to handle the flood of images, text and audiovisual content that constantly pour across the global networks.

Digital literacy is therefore one element in the i2010 Strategy's emphasis on Inclusion, better public services and quality of life. But this is not just about Inclusion –

ICT-related skills are vital for the competitiveness and innovation capability of the European economy.

In order to ensure internationally recognized computer skills certification in Europe, European Computer Driving Licence (ECDL) Foundation [4] was set up. ECDL provides international certification of computer skills and issues European Computer Driving License. ECDL Foundation's certification programmes have been delivered to over 10 million people, in 41 languages, across 148 countries, through a network of over 24,000 test centres. The certification programmes are designed to be accessible to all citizens, irrespective of age, gender, status, ability or race.

ECDL Foundation declares vendor neutrality [5]: "ECDL Foundation's certification programmes are vendor neutral. This means that our programmes are not tied to any one brand of software and are designed to enable people to gain skills that can be used in any appropriate software environment. This vendor neutrality gives people the freedom and flexibility to complete the training and testing, and demonstrate skills in a range of software applications."

However, complete vendor neutrality in computer skills testing process can be hardly observed. For instance, a project "Training in Computer and Internet Usage of Unemployed in Latvia" [6] implemented within the framework of European Community EQUAL by the Latvian Information and Communication Technologies Association (LIKTA), a member of ECDL Foundation issuing ECDL licences in Latvia, included training material „Basics for text processing". This material as recommended training material can be found on the Latvian ECDL certification homepage [7] includes Microsoft text processing tool Word as the only example of text processing environment. The situation remains the same in spreadsheet application training etc.

The open-source software products have an increasing market share in Europe and worldwide. For instance, the European portal www.webmasterpro.de published the results of its latest research [8] in February 2010. The study showed that in some EU countries market share of OpenOffice products reaches almost 22% (Poland, Czech Republic). In old EU member states, market share of OpenOffice products is: Germany – 21%, France – 19%.

This is another proof of increasing vendor neutrality in computer skills testing demanding adaptation of solutions for computer skills training and ECDL certification support tools to the changed market situation without causing unnecessary pressure and costs.

The research focuses on the following issues:

- Automated testing of computer skills used in a particular software environment;
- Applying the work assessment principles to the tests of computer skills;
- Testing the computer skills considering vendor neutrality principle.

2 Testing in Productive Environment

Bloom's Taxonomy [9] divides educational objectives into three "domains": Affective, Psychomotor, and Cognitive, each having domain specific skills to be developed.

Skills in the affective domain describe the way people react emotionally and their ability to feel another living thing's pain or joy. Affective objectives typically target the awareness and growth in attitudes, emotion, and feelings. Skills in the psychomotor domain describe the ability to physically manipulate a tool or instrument like a hand or a hammer. Psychomotor objectives usually focus on change and/or development in behaviour and/or skills.

Traditional education tends to emphasize the skills in the cognitive domain. These skills revolve around knowledge, comprehension, and critical thinking. There are six levels in the taxonomy, moving through the lowest order processes to the highest: knowledge, comprehension, application, analysis, synthesis, evaluation.

Thus, the knowledge is the very basic level of expressing skills – an exhibit memory of previously-learned materials by recalling facts, terms, basic concepts and answers. Developing skills at a higher level requires penetration, ability to analyze and evaluate and ability to apply the acquired knowledge in practice.

Computer skills refer to the level of cognitive domain application at least, since usage of a computer to the full extent is not possible without the ability to use specific applications. The process of learning computer skills including training is not possible without performing actions in certain environments. In such trainings, new knowledge is added to the existing knowledge of student through accomplishing tasks in practice.

Unfortunately, most tests of computer skills still involve testing techniques characteristic to the assessment of theoretical knowledge. Traditionally, it is organized as a test or exam where students provide answers to test questions in a free or structured or partly structured form. This approach, however, does not provide any significant proof to students' ability to use the particular environment. Moreover, this approach induces assumption that computer skills are derived from knowledge about computer skills that is generally not correct.

Testing of practical skills, on the other hand, shows student's ability to work with the particular application or in the particular environment where skills are learned. Controlling of these skills, as a rule, is difficult to automate since this would imply that testing software controls user's actions in a productive environment. This approach is called productive knowledge testing. Mostly recognized commercial product that supports this approach is EnlightKS [10].

Testing of practical skills includes various techniques and methodologies, and often combinations of them:

- Questions combined with images involves classic testing approach, however possible answers to test questions contain graphical information close to examples in real environment;
- Semi-practical tasks contain in advance prepared images of the environment where skills are to be tested. Correctness of the answers is determined by verifying relative or absolute position of computer mouse cursor in a given image against pre-defined mouse cursor position of the correct answer. Task images can be grouped or combined in series or presentations;
- Simulation of work environment is implemented by means of special software visually resembling the environment where skills are to be tested but being technically less sophisticated letting the user to perform limited set of actions

necessary to accomplish the task. Interactive videos are one of such possible ways of work environment simulation;

- Verification of work results is performed in a productive environment where specialised software solution verifies the result of performed actions against the correct answer;
- Assessment of user's actions is performed in a productive environment where specialised solution continuously verifies user's actions in order to determine not only correctness of these actions but also sequence and way how these actions were performed. Implementation of this approach requires very close integration between the testing software and the environment where the test is performed.

Verification of work results was considered as determinative testing methodology within the given research and experimental development since:

- the technical implementation is not restricted by a single version of productive environment – the test uses productive environment that is available on user's computer;
- the primary task of the test mechanism is to verify user's ability to perform the task, however, does not limit methodology or the way how the user reaches results within the given environment (it is acceptable not to perform the task in an optimal way);
- the automated verification of test results decreases the risk of human errors present in manually processed test results.

3 Implementation

The solution was implemented within the framework of experimental development applying findings of scientific research into practice. The developed prototype was approbated for two office desktop application packages – Microsoft Office and OpenOffice – mostly used Office software programs within the European Union.

The solution supports operations with different office desktop application packages, thus extending application area of the project outcomes, at the same time increasing complexity of technical aspects. Support is provided for OpenOffice version 3.2; Microsoft Office receives support for versions 2000 to 2007. Significant aspect of the project outcome – the solution supports also different operating systems such as MS Windows XP, MS Windows Vista and MS Windows 7. Enabling support for operating systems was one of the most sophisticated problems, mainly, due to different security settings and other technical nuances.

3.1 Methodology

Implementation of solution is based on generalized automated testing methodology design for universal application, respectively, independent from office software vendor. Moreover, a self-evident request was the independence from different product versions from the same vendor.

The main methodological task was to define a set of simple, measurable, separable and autonomous units that would cover the area of testable knowledge and skills.

Knowledge and skills were divided into smaller units. Each such unit includes a small set of knowledge and skills that undergo no further detailing. Each of these units can be used separately or in combination for testing and training purposes accordingly to the difficulty level and knowledge area.

It was important that the standard specification for computer skills of an office desktop application without specifying a particular product is already considered in the design phase. Thus, the solution had to function equally well and independently from the chosen application.

Such an approach complies with the requirement to provide a possibility to design test tasks without taking interest in which environment tests will be performed. Students would receive equivalent evaluation independently of the environment they took test in.

During the research process, several potential problems and ambiguously interpretable opportunities were identified. These problems mainly comprised cases when skills for different products or different product versions have to be tested by different methods.

Purposely not restricting probable outcomes of the research each probable problem situation was evaluated separately – eventually some of them were excluded from test unit set, but, however, some were implemented. As a result, the developed testing software provides more features to end users, though a partial responsibility for the outcome lies on the person who defines the tasks. Within the development phase of solutions, test units were described and developed. With the help of these test units, sets of practical tasks for testing computer skills were defined. Each of these units comprises logically simple tests, and each unit corresponds to a specific ECDL requirement, e.g., a skill to apply a specific font to text or to change its colour or size.

3.2 Technical platform

Since office applications are usually made as desktop applications, the productive testing environment should also have a form of a desktop application.

The development was performed in VisualBasic.Net for Windows environment. Good compatibility with MS Office applications was a strong argument to choose this development environment. However, this methodology can be successfully implemented also with other application development tools.

Each of the test units was developed as a separate VisualBasic.Net function whose tasks included test object and its properties to be checked. The system returns true, if the property named by the argument exists and its value matches one defined as function's argument, and false, if there is no property or property value does not match the defined desirable outcome. Specification, development and testing of such test units comprised a very significant portion of project time resources.

Considering autonomic nature of each unit within the given implementation, unit testing method was used to test software units. This method provides comprehensive overview of status of completion and quality of developed skills evaluation units.

3.3 Architecture

Although the selected solution is not complex and it has homogeneous implementation logic, the practical implementation involved several technically sophisticated issues.

Data gathering in format that would enable testing of unit logic presented the most significant problem during implementation.

For instance, if logical test *IsFont* includes simple test or an element with a specific font, then the implementation should include testing the object itself and accessing its components and properties. In a particular case, if the test would imply a MS Word document as a result, then in order to check fonts in a part of the document, the test mechanism should know how to access this part of the document and its properties.

The above-described access to objects has been implemented by means of interpreter. The interpreter has three interconnected components:

- Data model – a structure where processed document data are filled;
- Application interface – a subsystem providing data gathering and usage;
- Test units – the implementation of the logic of the test.

Architecture of the interpreter is shown in Figure 1.

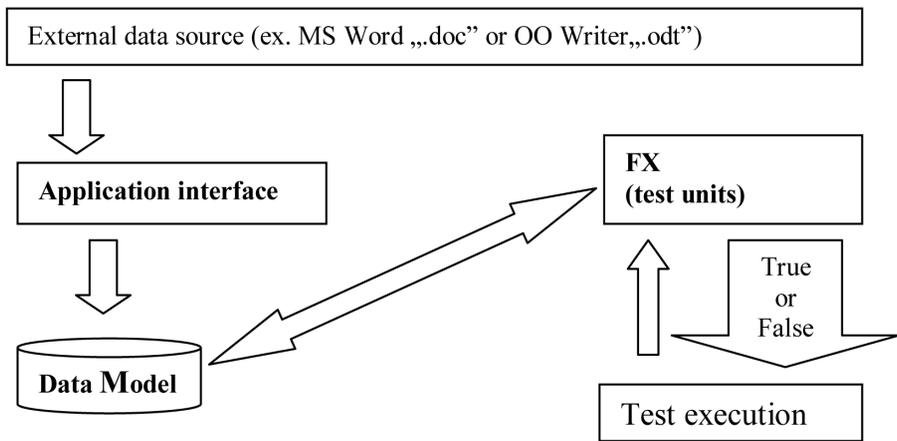


Fig. 1 Architecture of interpreter

Data model provides a possibility to store information needed for testing logic of units. Moreover, data structures are general by nature, namely, independent from source and manner of use.

Data loading is organized via help of application programming interface (API) classes. Initially, general classes were defined for description of general interface. Further, significantly different API classes were implemented for specific applications in MS Office and OpenOffice.

The component of test units derives values form the data model and executes the test logic in units.

This kind of architecture enables complementing the API without significant changes in data model and testing units in order to add support for other office software products in the future.

4 Implementation of Application Programming Interface

Due to specifically different applications, the development of application interface was the most sophisticated part in the implementation of testing interpreter.

Implementations were different in both cases, MS Office and OpenOffice:

- Significant discrepancies in applications' measurement units;
- Even within one program package such operations as program start and storage of results were different in each application. For instance, Open Office regards diagram saved in MS PowerPoint format as a picture. However, a diagram created in OpenOffice is not recognized by MS PowerPoint at all;
- Different conception in other significant matters, for instance, MS Word and OpenOffice Writer, have different algorithms for word counting in a document.

Next chapters deal with comparative overview of technical implementation.

4.1 Processing methods for MS Office objects

For processing MS Office documents Microsoft recommends using Primary Interop Assemblies [11].

A primary interop assembly is a unique, vendor-supplied assembly that contains type definitions (as metadata) of types implemented with COM. There can be only one primary interop assembly, which must be signed with a strong name by the publisher of the COM type library. A single primary interop assembly can wrap more than one version of the same type library.

A COM type library that is imported as an assembly and signed by someone other than the publisher of the original type library cannot be a primary interop assembly. Only the publisher of a type library can produce a true primary interop assembly, which becomes the unit of official type definitions for interoperating with the underlying COM types.

Publishers of COM components produce primary interop assemblies and distribute them to developers for use in .NET Framework applications.

To use the features of a Microsoft Office application from an Office project, the primary interop assembly (PIA) must be used for the application. The PIA enables managed code to interact with a Microsoft Office application's COM-based object model [12].

4.2 Processing methods for OpenOffice objects

UNO Project provides processing of OpenOffice documents [13]. UNO stands for Universal Network Objects. UNO is one of the accepted projects of OpenOffice.org.

UNO is the component model of OpenOffice.org. UNO offers interoperability between programming languages, other component models and hardware architectures, either in process or beyond process boundaries, in the Intranet as well as in the Internet. UNO components may be implemented in and accessed from any programming language for which a UNO implementation (AKA language binding) and an appropriate bridge or adapter exists.

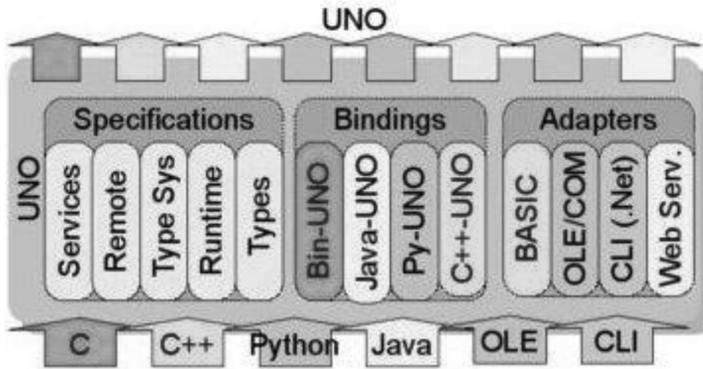


Fig. 2 OpenOffice possible usages using UNO project

4.3 Comparative evaluation

The implementation process contained a sophisticated challenge to overcome differences among PIA and UNO in order to develop solution according to specification. Moreover, most UNO documentation is written for Java programming language. Samples for usage with other languages were scarce or there were hardly any.

Within last years, OpenOffice has brought in several changes in object structure by extending UNO features, thus facilitating access to object information. However, these changes caused ambiguity regarding the volume of available information, namely, the documentation is not brought up-to-date and huge volume of available information still refers to outdated interface versions, thus hindering the research efforts.

By using PIA, it is possible to access different object properties and data units easily. However, PIA enables working with „com” [14] objects only; therefore, it does not provide a possibility to review „com” object errors in problem cases.

UNO has a different framework, however its discrepancies with PIA due to complex access to data details are only perplexing development. Moreover, in comparison to PIA, in UNO, the data set requires additional processing efforts in order to obtain the needed data. In the case of PIA, it was comparatively easy to obtain object’s properties and process its values. In the case of UNO, additional efforts were required to both identification of the needed property among others in the set and obtaining the needed value form all available values of a property.

Some difficulties were caused by the necessity to use proper data objects in order to access further properties. False objects did not have appropriate properties, but finding the proper objects was hindered by the scarce and incomplete documentation.

5 Results and Conclusions

The ideas developed during the project implementation were transformed into prototype, which was approbated for testing purposes of computer skills in office software environments of two different vendors.

During development phase, 140 units of skill testing were specified. Implementation was accomplished for 120 such units. Due to various logical inconsistencies between office software from two different vendors discovered during the prototype approbation phase, an implementation of an interpreter (a practical implementation of the prototype design) could not be done equally for different office software.

The results of the project are both ideological and technological by nature.

Benefits of automated testing of computer skills in productive environment are the following:

- Optimized testing process of computer skills – significant reduction in time needed for testing and gathering the results, significant reduction (in most cases even total absence!) of evaluation errors. Unified and neutral attitude towards every student is ensured;
- Complete identity between software product versions used in test and as a working environment is endured – if the student is able to perform the action within the skill testing environment, then it can be asserted that she/he will be able to master these skills in a real working environment as well;
- Accumulated data base of computer skill testing – tasks, examples, good practice – that can be used in long term period and provide good basis for the improvement of teaching and training methodology;
- Automatically accumulated valuable statistics – valuable data on tests, test results both on general and detailed level (e.g. dynamics of a student's development) can be obtained without significant efforts and additional resources;
- Computer skill testing materials (sets of tasks, etc.) can be used for training purposes as long as solution can technically „replay” testing process step-by-step.

However, a few restrictions and challenges for the use of solution exist:

- The leading software applications are subjects to permanent improvement. Although the newest product versions are supposed to be compatible with the previous ones, technical implementation of the newer versions can significantly differ from the previous versions, thus causing necessity to continuously maintain automated testing solution's consistency with market development;
- In order to comply with the principle of vendor neutrality, testing the computer skills would have to use only generalized and standardized cluster of features characteristic to most environments, where skills are tested;
- Generally, every environment eligible for testing skills needs a technically individual solution in order to automatically verify students' test results against correct answers.

Within the framework of this research project, the following objectives were accomplished to a full extent:

1. Specification of testing skill units;
2. Development and prototypes of interpreters;
3. Testing the correct operation of a prototype during its testing phase;
4. Approbation and testing the end version of interpreter prototype.

6 Acknowledgement

The research is supported by the European Regional Development Fund (ERDF).

References

1. http://ec.europa.eu/education/archive/elearning/index_en.html
2. http://ec.europa.eu/information_society/tl/edutra/index_en.htm
3. Reding, V.: i2010: A new start for Lisbon and for European Information Society and Media policies. The i2010 Conference, London, 6 September 2005
4. <http://www.ecdl.org>
5. <http://www.ecdl.org/index.jsp?p=94&n=928>
6. <http://www.latvijapasaule.lv/>
7. <http://www.ecdl.lv>
8. <http://www.webmasterpro.de/portal/news/2010/02/05/international-openoffice-market-shares.html>
9. Bloom, B., Englehart, M. D., Furst, E. J., Hill, W. H., Krathwohl D. : The Taxonomy of Educational Objectives, The Classification of Educational Goals, Handbook I: Cognitive Domain, 1958
10. http://www.knowledgeuk.com/index.php/main/show/0/5/Home_Page
11. <http://msdn.microsoft.com/en-us/library/aax7sdch.aspx>
12. <http://msdn.microsoft.com/en-us/library/15s06t57.aspx>
13. <http://udk.openoffice.org/>
14. http://en.wikipedia.org/wiki/Component_Object_Model