

# Evaluation of TENCompetence proof of concept assessment tools

Citation for published version (APA):

Petrov, M., Aleksieva-Petrova, A., Stefanov, K., Schoonenboom, J., & Miao, Y. (2008). Evaluation of TENCompetence proof of concept assessment tools. *Default journal*.

## Document status and date:

Published: 08/09/2008

## Document Version:

Peer reviewed version

## Document license:

CC BY-SA

## Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

## General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

<https://www.ou.nl/taverne-agreement>

## Take down policy

If you believe that this document breaches copyright please contact us at:

[pure-support@ou.nl](mailto:pure-support@ou.nl)

providing details and we will investigate your claim.

Downloaded from <https://research.ou.nl/> on date: 06 Dec. 2024

Open Universiteit  
[www.ou.nl](http://www.ou.nl)



# Evaluation of TENCompetence proof of concept assessment tools

**Milen Petrov, Adelina Aleksieva-Petrova, Krassen Stefanov**

Faculty of Mathematics and Informatics,  
University of Sofia, Sofia, Bulgaria,  
E-mail: {milenp, adelina, krassen}@fmi.uni-sofia.bg

**Judith Schoonenboom**

SCO-Kohnstamm Institute,  
University of Amsterdam, Amsterdam, Netherlands  
E-mail: J.I.Schoonenboom@uva.nl

**Yongwu Miao**

Educational Technology Expertise Center,  
Open University of The Netherlands, Heerlen, Netherlands  
E-mail: yongwu.miao@ou.nl

## Abstract

*The TenCompetence Assessment Model was developed as an attempt to develop complex but feasible for real implementation assessment model, corresponding to the last achievements in this field [6]. Therefore the Proof of Concept assessment tools are very important, aiming to test the balance between TENCompetence framework, technology, assessment model, target audience and user acceptance, addressing mainly the feasibility of the TENCompetence Assessment Model. The Proof of Concept assessment tools are not looking for a single end-to-end solution but for the creation of set of mini-assessment environments, sharing common Assessment Model, in which key elements and their dependencies can be tested and verified.*

**Keywords:** evaluation report, ISO 9241, TENcompetence assessment model, learning technologies

## 1. Introduction

Within the framework of TENCompetence project [1] there was developed new assessment model suitable for lifelong competence development [6]. This model is rather complex, or meta-model for describing different types of assessments and helping to model some assessment processes.

Our main goal was to prove that this model will work in practice. For this purpose we need to implement the model developing different types of assessments. We choose to implement two different assessment approaches using the model - 360 degree feedback and portfolio assessment. For this purpose we implemented two proof-of-concept tools: assessment editor tool and run-time assessment tool (helping the end user to

apply the model in order to implement the chosen assessment examples).

In this paper we describe the evaluation of these tools. During the evaluation we aim to answer the following questions: (1) are the tools capable for the implementation of the TENCompetence Assessment model, and (2) can we apply the model in order to perform the two chosen assessment methods.

The evaluation took place in the period between 1st of September and 31st of October 2007. In order to answer the two above stated questions, we develop test scenarios and test cases. They serve as the main requirements for the usability and feasibility of the model and the proof-of-concept tools. The aim of the evaluation was to measure the extent to which the tools met their aim (matching to requirements)

and to test the feasibility of number of key components of the TENCompetence Assessment Model (described formally and known also as Simple Assessment Specification v. 1.0, see [6]).

The proof-of-concept assessment tools have the following main functionalities: Assessment Authoring and Runtime tools (fig. 1).

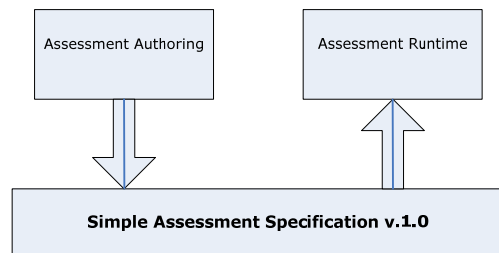


Figure 1: Functionality of tools

The tools consist of the following components:

1. Assessment Editor, where assessment architect can define the blueprint of the assessment or model the assessment process during the assessment design phase.

2. Question editor, used to create and edit different type of questions, structured in three main categories: demonstration items, construction items, and selection items (the assessment construction phase).

3. Test Editor, combining the model defined with the Assessment Editor with the assessment items, defined with the Question editor, resulting in the run-time scheme.

4. Assessment run-time simulator – used to perform the evaluation by executing the run-time scheme.

5. Response processing (partly implemented) – for saving and tracking the results from the assessment performed.

## 2. Methodology and instruments

In order to validate the new assessment model developed, as well as the feasibility of the proof-of-concept tools implemented, we apply two standard evaluation methods.

The first one was used to evaluate how the tools correspond to the functional requirements, using the specially developed

review checklist (developer checklist). The second one was used to rate the interface and usability of the systems, using end user questionnaires. As the functional requirements are the most important issue (aiming to be used to find answers to the main objectives for the evaluation), we pay more attention to it, as this was the principal goal of our evaluation.

The evaluation of the quantity and quality of the software was not the focus of this evaluation, as the purpose of the applications concerned is to provide proof of concept for the TENCompetence Assessment Model. Nevertheless, this question also has indirect impact to the functional requirements, so we pay attention to it as well. We provided the source code to two evaluators who carried out an expert inspection.

### 2.1. Developer review checklist

As the tools were used to implement two different assessment methods - e-portfolio assessment and 360 degree feedback, two different checklists were prepared:

- check list for portfolio example
- checklist for 360 degree feedback example

The functional testing of the tools was based on the Test Scenarios and Test Cases, prepared using the Use cases and scenarios available. For each Test Case we defined Test case document, which needed to be filled in during the testing. In this document the evaluator recorded the result from each step (pass/fail), all errors encountered, as well as all problems and observations in the special Notes field.

For each test case there was also test scenario available (as a separate pdf document), describing in details what exactly the evaluator has to perform (what tool to download and from where, what task to perform with this tool, what results to achieve and record, etc. The main results from evaluation, as described in the filled test case document, were used for the analysis of the tool evaluation.

## 2.2 User questionnaire

The purpose of this instrument was to assess the usability of the tools. It does so by asking end users a number of questions regarding how easy the tool can be used. This questionnaire was structured in two parts:

- General information about the user and his/her experiences with using information technologies.
- Evaluation of the tools adapted from some principles like suitability of the system, controllability and conformity with user expectations.

The questionnaire comprises 29 items adapted from the six design principles of ISO 9241 (Part 10) which provides information that can be used within iterative software development [2]:

- suitability to task

1	The way in which data is entered is suited to the tasks I want to perform with the software	1 - 2 - 3 - 4 - 5
2	Too many different steps need to be performed to deal with a given task.	1 - 2 - 3 - 4 - 5
3	In a given screen, I find all of the information I need in that situation.	1 - 2 - 3 - 4 - 5
4	The terminology used in the software reflects that of my work environment.	1 - 2 - 3 - 4 - 5
5	I can easily adapt the software for performing new tasks.	1 - 2 - 3 - 4 - 5
6	I understand immediately what is meant by the messages displayed by the software	1 - 2 - 3 - 4 - 5
7	It is easy to retrieve information about a certain entry field.	1 - 2 - 3 - 4 - 5
8	The software provides me with enough information about which entries are permitted in a particular situation.	1 - 2 - 3 - 4 - 5

Figure 2: Part of the given user questionnaire

## 3. Evaluation process

### 3.1. Procedure

Detailed evaluation instructions in the form of user guide were written and given to evaluators (test scenario). The evaluators followed the evaluation instructions and executed the following evaluation steps:

Step 1: Download evaluation bundle which contains the user guide and pre-configured self-assessment instruments (test case, questionnaires, checklists, etc.)

Step 2: Download the corresponding proof-of-concept tool (precompiled version built for evaluation).

Step 3: Unpackage and install the tool.

Step 4: Work with the tool, fill in the results and return the test documents.

- self explanatory
- controllability
- conformity with user expectations
- error tolerance
- suitability for individualisation

The statement of each item is assessed on a five rating scale starting from 1 (“predominantly disagree”) to 5 (“predominantly agree”). A further option (“no option”) is offered to reduce arbitrary answers.

We used groups of students in software engineering in their first year of study, in order to make self-assessment and position them in course “Programming fundamentals”. They have a variety of backgrounds.

In the figure below shown the part of the given user questionnaire (fig.2):

Step 5: Fill in the feedback form (end user questionnaire) for the evaluation of the tool.

Step 6: Return the filled feedback form.

### 3.2. Collection and analysis of the results

This step included the collection of all the data (filled in tables, checklists and questionnaires) from the evaluation of the tools and making summary and analysis of these results.

The report should also include gap analysis by providing a general statement of the capability of the system as demonstrated by the test case, compared with the requirements, stating the system deficiencies and recommending improvements of the system.

The evaluation was carried out with 40 people. They received the User Guide, representing particular evaluation process, in Bulgarian [3]. They also receive pre-assessment, which is not a traditional assessment, but rather a self-assessment of competences.

Table 1: Details of evaluation process

Description	Number of users	Note
Software Unique users downloads	52 users	3 – test users
unique users/after removing test-users and anonymous users	48 users + 1 anonymous	1 anonymous
Software non-unique users	73 Downloads	Software downloads and other items from site
Returned answer from assessment	33 users	
Returned feedback form	20 users	3-invalid; 2-blank
Returned valid forms	15 forms	

In the table below provides details of the evaluation process:

The completed questionnaires, tests, test summary, and test analysis.

#### 4. Analysis of the Evaluation Results

The results from the Functional testing show that almost all basic functionalities were fully or partially implemented, following the use cases and scenarios developed. There were few items which were not implemented, but they were not critical for the overall use of the tools. For example, in the Item Construction phase of the Assessment Model, the demonstration and construction items were not implemented. The list of the principal functionalities implemented includes:

- Create an assessment plan
- Delete an assessment plan
- Edit an assessment plan
- Create an item
- Delete an item
- Edit an item
- Browse items
- etc.

The results of the questionnaire for the six design principles are summarized in the table 2 below, and provide the average response results.

The average response is close to 3,5 points which indicates that the level of the user satisfaction is rather high.

Table 2: Summarised results

Design principles of ISO 9241	Average result
Suitability to task	3,69
Self explanatory	3,64
Controllability	3,79
Conformity to user expectations	3,08
Error tolerance	3,33
Suitability for individualisation	3,62

#### 5. Conclusions

Overall conclusions regarding the tools are:

- Responses for the six design principals targeted in this evaluation were relatively high, with an average close to 3.5 and with the scores for all principals being higher than 3. Although this questionnaire does not go into details of the interface, the result suggests that the tools have (at the least) a reasonable standard of usability and quality.
- In formative testing during the development process several

interoperability issues were identified, with an incompatibility between the Assessment Specification and the target output formats (IMS LD and QTI) [4,5]. This incompatibility meant that it was neither possible nor necessary to evaluate this aspect of the tools.

- The editor and player were stable, although some interoperability issues between them were identified
- Testing against various small sub-components of the assessment models, indicates that the specification has good modelling power
- Expert inspection indicated that several parts from the tool interface were not sufficiently human-friendly, and that while the tools are usable, the interface is still too close to the specification
- The specification is based on xml and the documents produced are hard for a human to read.

On the base of the analysis of the evaluation results, it was proposed to reduce the learning curve for using the tools with the development of usage profiles. These profiles could be used for the proper user interface design and development, which will make the use of the tools more natural and user friendly. The candidate profiles are identified below:

1. Profile for user “assessment process designer” (or assessment stakeholder, which provides blueprint for assessment process)
2. Profile for domain expert of assessed assets (assets like knowledge, skills, traits). Under “domain expert” we recognize person which is assessment item author/editor/designer/selector.

3. Profile for assessment developer (selects items from item bank, provided by domain expert; configure required fields in order to make assessment “runnable” or “do-able”)
4. Profile for peer/self/360degree evaluator (person which run/evaluate configured assessment)
5. Profile for response rater (this profile is only identified, but is out-of-scope for current research, and developed tools). This can be target for further research.
6. Profile for decision maker (as profile in point 5 - this profile is also only identified, but is out-of-scope for current research, and developed tools). This can be target for further research.
7. Overall profile/profile for “super-admin” or “super-consultant”, which is combination of all profiles, described from (1) to (6). This profile can help in checking the assessment model and assessment process for integrity.

## References

- [1] Koper, R. & Specht, M. (2007). TenCompetence: Lifelong Competence Development and Learning. In: Sicilia, M.-A. (Ed.). *Competencies in Organizational E-Learning: Concepts and Tools*. Idea Group.
- [2] ISO 9241-110, Ergonomics of human-system interaction -- Part 110: Dialogue principles, 2006
- [3] Aleksieva-Petrova, Adelina, and Petrov, Milen. Assessment proof-of-concept tools user guide, ver.1.0. Retrieved April 2008 from: <http://dspace.ou.nl/handle/1820/1110>
- [4] IMS LD, Retrieved May 2007 from: <http://www.imsglobal.org/learningdesign/>
- [5] IMS QTI Retrieved May 10, 2007, from: <http://www.imsglobal.org/question/>
- [6] TENCompetence Deliverable D6.1 – ANNEX 2 – Assessment, Retrieved April 2008 from <http://dspace.ou.nl/handle/1820/1150>