Towards a UMLsec-Based Proctored Examination Model

Ibukun Fadahunsi, Federal University of Agriculture, Abeokuta, Nigeria
Oluwasefunmi ‘Tale Arogundade, Federal University of Agriculture, Abeokuta, Nigeria
Adesina Sodiya, Federal University of Agriculture, Abeokuta, Nigeria
Bakai Olajuwon, Federal University of Agriculture, Abeokuta, Nigeria

ABSTRACT

Electronic examination systems are becoming increasingly complex and intensive to develop with the introduction of virtual invigilator in proctored examinations. In order to address this complexity issue, there is evident need to have a global model that is extensible. Modeling software systems enables developers to better understand the system they are building and offers opportunities for simplification and reuse. This article presents an extension to UMLsec, by introducing three (3) new stereotypes which were added to the UMLsec Profile for a Proctored e-Exam model. The model was validated and converted to a platform specific model using the Java stereotype available on Papyrus. This enabled the model generate Java classes which can be used for the implementation of a secure proctored e-exam system. The model allows developers with little or no knowledge in security to use the model to build proctored e-exam systems and to incorporate all known security requirements. The model can also be extended to accommodate new security solutions for e-exam systems as they are discovered.

KEYWORDS

INTRODUCTION

Electronic examination systems are fast becoming the major platform for assessment used by Massive Open Online Courses (MOOC) providers, vocational and professional certification bodies and school exam boards. It is preferred to the traditional method of assessments due to numerous benefits it offers among which are: - automated marking, on-demand tests, immediate feedback, randomization of questions, integrated assessment activities with tasks based on videos assessment, text assessment, etc. (Kikelomo, Gray, & David, 2010).

With the adoption of the online examination, accreditors and learning organizations are becoming increasingly concerned with threats poised on e-exam systems. Questions such as who is actually taking the course behind the screen and whether students are taking examinations without benefit of accessing unauthorized resources over the internet are frequently asked. Even if instructors assume that students are not using textbook during examinations, what about other resources not allowed by in-class students, such as cell phones, Google searches, the Wolfram Alpha answer engine, and additional notes? (Cochran, Troboy, & Cole, 2010). Security issues facing online exam delivery is on the increase amongst which are challenges of user identification, integrity violation, privacy issues, breach of trust, confidentiality issues, masquerading attacks, cheating, ensuring continued presence of student all through the exam session, communication channels breakdown, etc. (Guttorm & Aparna, 2019).

With advancement of technology, proctored e-exam system has been introduced to the e-learning community to address some of these security issues. A proctored e-exam system is a type of e-exam system that employs the use of proctors (software or human) to monitor students taking examinations from a remote location using webcam and other audio devices throughout the online exam session. During this process the actions of the exam takers are observed to identify any cheating action from learners. A proctor is a person appointed to keep watch over students during an examination. It could be human or software. A software proctor is a computer application programmed to monitor student activities during an exam through storage of video streams of the exam session for later review by the exam authority or teachers.

With the introduction of virtual invigilator and the use of proctored examinations, the system is becoming increasingly complex and mission critical. In order to address this complexity issue, e-exam system needs to be modeled. “Modeling is the designing of software applications before coding. It is an essential part of large software projects, and helpful to medium and small projects as well. Using a model, those responsible for a software development project’s success can assure themselves that business functionality is complete and correct, end-user needs are met, and program design supports requirements for scalability, robustness, security, extendibility and other characteristics, before implementation in code renders changes difficult and expensive to make” (OMG, 2005). Hence, the authors propose to use the Unified Modeling Language (UML) to develop a secured e-exam model due to its extendable features and its industry wide acceptance. Most developers are trained to develop systems
Exploring the Impact of Learning Styles on the Acceptance of Open Learner Models in Collaborative Learning

Heterogeneous Mobile Computing Issues
[www.igi-global.com/chapter/heterogeneous-mobile-computing-issues/121257?camid=4v1a](www.igi-global.com/chapter/heterogeneous-mobile-computing-issues/121257?camid=4v1a)

Implementing Internal Software Process Assessment: An Experience at a Mid-Size IT Company
[www.igi-global.com/chapter/implementing-internal-software-process-assessment/61211?camid=4v1a](www.igi-global.com/chapter/implementing-internal-software-process-assessment/61211?camid=4v1a)