

EVALUAREA CUNOȘTINTELOR BAZATĂ PE O ONTOLOGIE

Radu Ciora, drd., *Universitatea Lucian Blaga
din Sibiu*

Carmen Simion, prof.dr., *Universitatea Lucian
Blaga din Sibiu*

Claudiu Voinia, drd., *Universitatea Lucian
Blaga din Sibiu*

ONTOLOGY DRIVEN ASSESSMENT

Radu Ciora, *Ph.D. Student, Lucian Blaga
University of Sibiu*

Carmen Simion, *Prof.Dr., Lucian Blaga
University of Sibiu*

Claudiu Voinia, *Ph.D. Student, Lucian
Blaga University of Sibiu*

REZUMAT - Evaluarea cunoștințelor elevului este ultimul pas al ciclului de învățământ. Vă prezentăm o abordare nouă a verificării cunoștințelor pe calculator. Implementarea prototipului oferă mijloace de evaluare inteligente, conștiente de mediul de lucru. Sistemul demonstrativ este un sistem de monitorizare bazat pe Windows și condus printr-o ontologie. Elevului i se atribuie o sarcină exactă, iar sistemul nostru analizează acțiunile pe care elevul le efectuează și le compară cu o listă de acțiuni care trebuie executate, pentru ca această sarcină să fie îndeplinită cu succes. Dacă toate acțiunile sunt executate în ordinea corectă, atunci spunem că elevul a finalizat cu succes sarcina. În cazul în care elevul nu reușește să termine o activitate, el va primi un feedback, care specifică pașii pe care acesta i-a greșit.

Cuvinte cheie: ontologie, evaluarea cunoștințelor, e-learning

I Introducere

Sistemul nostru este o aplicație de evaluare a cunoștințelor bazată pe o ontologie. Acesta este capabil de a evalua un elev, fără interacțiune umană. Acest lucru este realizat cu ajutorul sistemului de raționament bazat pe o ontologie RDF/RDFS [1] care utilizează SPARQL [15] ca limbaj de interogare. Până acum, evaluarea elevului de era făcută utilizând chestionare cu răspunsuri multiple sau erau bazată pe răspunsul la întrebări [4]. Sistemul nostru este în măsură să evalueze un elev în timp ce el execută o anumită sarcină dată. În funcție de acțiunile pe care elevul le întreprinde, sistemul decide dacă el a terminat cu succes sarcina sau nu a reușit acest lucru. Prin urmare, sistemul nostru ușurează munca examinatorilor, îmbunătățind astfel gestionarea timpului în clasă, lăsând mai mult timp pentru alte activități.

ABSTRACT — Learner knowledge assessment is the final step of the educational cycle. We present a novel approach to computer-based knowledge verification. The prototype implementation provides the means of assessment in a context-aware e-learning environment. The demonstration system is an ontology-driven Windows-based monitoring system. The learner has to perform a specific task and the computer analyzes the actions that he performs and compares them against an action list that needs to be fulfilled for that particular task to be fulfilled. If all the actions are executed in the necessary order, then we say that the learner has successfully completed the task. If the learner fails to complete a task, he will be provided with feedback, specifying the steps where he has failed.

Key words: ontology, knowledge assessment, e-learning

I INTRODUCTION

Our system is an ontology-based e-learning assessment application. It is able to assess a learner, without human interaction. This is achieved with the aid of reasoning system based on an RDF/RDFS [1] ontology that uses SPARQL [15] as a query language. Until now, learner assessment was made by using multiple choice questionnaires or was question-answering based [4]. Our system is able to assess a learner while he performs a specific task that he receives. Based on the actions that the learner performs, the system decides whether he successfully completed the task, or if he failed to do so. Therefore, our system reduces the number of tasks that supervisors has to perform and it improves the time management of classes, leaving more time for other activities. The core of the

Sistemului are la bază o ontologie conștientă de context [11]. Pentru a fi în măsură să verifice acțiunile celor care învață ", vom folosi interfața COM [12] pentru comunicarea cu aplicațiile și API-ul interfeței cu utilizatorul pentru a obține numele elementelor acesteia. Sistemul nostru monitorizează elevul, înregistrează acțiunile sale și le compară cu o listă de acțiuni prestabilă. În cazul în care acțiunile înregistrate se identifică cu ceea ce se așteptă ca rezultat, spunem că testul a fost trecut, altfel vom spune că sarcina nu a fost finalizată cu succes, și se prezintă elevului acțiunile pe care acesta le-a greșit și totodată un mod cum ar fi putut duce la bun sfârșit sarcina.

II Contextul cercetării

Cercetarea din cadrul proiectului ASAP [8] a generat un set de instrumente pentru evaluarea elevilor. Acesta a constatat în "Întrebări generate de răspunsul studentului" și "Întrebări generate automat". "Întrebările generate de răspunsul studentului", sunt evaluate folosind un sistem automatizat de notare în prezent este folosit un XML simplu [5] pentru protocolul de răspuns. Sistemul de notare se aseamănă cu o testare modulară. Serviciul web "Roboproof" generează întrebări cu un număr aleator, dar controlat de modificări.

Cercetarea din cadrul proiectului JELFAD (JISC – sistem demonstrativ de evaluare în e-learning) [9] încearcă să demonstreze funcționarea instrumentelor și conținutului din cadrul platformei lor de e-learning ce pot fi evaluate într-un context activ de evaluare a studenților. Un al doilea scop al acestui proiect a fost de a evalua succesul cu care ieșirile din succesiunea sau proiectele de învățare pot fi integrate cu rezultatele din proiectele de evaluare. Al treilea obiectiv al acestui proiect a fost de a furniza date și feedback pentru FREMA [6] – modelul de referință al evaluării platformei de e-Learning.

Proiectul TReCX (urmărirea și raportarea în contexte ale e-learning-ului) [10] este un set de instrumente „open source”, care este capabil de a urmări elevii în sisteme distribuite de e-learning. Acesta permite

system is based on a context-aware [11] ontology. In order to be able to check the learners' actions, we use the COM [12] interface for communicating with the applications and the user interface API for obtaining the name of the UI elements. Our system monitors the learner, records his actions and compares them against a list of determined actions. If the recorded actions match the expectation list, then we recognise the test as successful, otherwise we say that the task was not successfully completed, and we show the learner the actions that he did wrong and a way how he could have completed the task successfully.

II BACKGROUND AND CONTEXT

The existing work from the ASAP project (Automated System for the Assessment of Programming) [8] produced a set of tools for student assessment. It consisted of the ‘Student Generated Response Questions’ and the ‘Automatically Generated Questions’. The ‘Student Generated Response Questions’ are evaluated using an automated marking system currently using a simple XML [5] response protocol. The marking system corresponds to a unit testing approach. The ‘Roboproof’ Web Service generates questions with a controlled random number of modifications.

The work of the JELFAD project (JISC e-Learning Framework Assessment Demonstrator) [9] tries to demonstrate the operation of their e-Learning Framework tools and content that could be evaluated in a live student assessment context. A second goal of this project was to evaluate the success with which outputs from sequencing or learning projects can be integrated with the outputs from the assessment projects. A third aim of this project was to provide data and feedback to the FREMA [6] eLearning Framework Assessment Reference Model.

The TReCX project (Tracking and Reporting in e-Learning Contexts) [10] is an open source toolkit that is able to perform

generarea și colectarea datelor de urmărire, de la o varietate de aplicații e-learning și apoi prezentate ca un tot coerent, astfel încât acțiunile unui cursant să poată fi urmărite și dincolo de granițele aplicației. TreCX este site web de sine stătător ce conține o baza de date ce poate stoca evenimente, care pot fi utilizat de aplicațiile de e-learning pentru a stoca acțiunile celor care învață într-un format XML. Acesta conține și o interfață de publicare care permite unei aplicații existente să trimită evenimente în baza de date, folosind o interfață REST. Există și un modul de raportare, cu o interfață REST pentru a facilita crearea de aplicații de raportare care pot interoga una sau mai multe instanțe ale bazei de evenimente.

III Problema

Până acum, instrumente de evaluare nu furnizau un mecanism de verificare în practică a cunoștințelor. Nu au existat instrumente automate de evaluare în măsură să evalueze un elev în timp ce el execută o anumită sarcină pe un calculator. Soluțiile existente de evaluare disponibile pe piață erau chestionare cu variante multiple, care pot conține un anumit grad de dezordine controlată în întrebările generate.

De exemplu, să luăm cazul unui elev care este trebuit să trimită un e-mail la „Foo.Bar@ulbsibiu.ro”, utilizând Microsoft Outlook, cu subiectul: „e-mail de test” și un conținut gol. Într-un chestionar am fi avut următoarea întrebare:

„Care dintre următoarele acțiuni sunt obligatorii pentru ca un utilizator să trimită un e-mail:

- a) Porniți Microsoft Outlook
- b) Faceți clic pe butonul New e-mail
- c) Completați câmpul To
- d) Completați câmpul Cc
- e) Completați în câmpul Subject
- f) Scrieți textul e-mail
- g) Apăsați pe butonul Send”.

Răspunsurile corecte ar fi a, b, c și g. Dar ce se întâmplă dacă doriți să testați practic dacă elevul știe cum să trimită un e-mail. În mod

tracking of learners across distributed e-learning systems. It allows tracking data to be generated and collected from a variety of e-learning applications and then presented back as a coherent whole so that the actions of a learner can be followed across application boundaries. TreCX contains a stand alone web-based tracking event store that can be used by e-learning applications to store the actions of learners in an XML format; a publishing library that enables an existing application to push events to the tracking store using a REST interface. A reporting module with a REST interface exists for facilitating the creation of reporting applications that can interrogate one or more tracking store instances.

III PROBLEM

Until now, assessment tools did not provide a mechanism for hands-on knowledge testing. There were no automated assessment tools able to assess a learner while he performs a specific task on a computer. The existing assessment solutions available out on the market were multiple choice questionnaires, which may contain some degree of controlled randomness in the generated questions.

For example, let’s take the case of a learner who is requested to send an e-mail to ‘Foo.Bar@ulbsibiu.ro’, using Microsoft Outlook, with the subject: ‘test e-mail’ and an empty body. With a questionnaire you would have a question like the following:

Which of the following actions is a user required to do in order to send an e-mail:

- Start Microsoft Outlook
- Click the new e-mail button
- Fill the To field
- Fill the Cc field
- Fill the Subject field
- Write the e-mail text
- Click the send button

The correct answers would be a, b, c & g. But what if you want to test whether the learner knows how to send an e-mail or not.

obișnuit pentru acest caz va trebui să puneti elevul în fața unui calculator și să aibă un profesor care să-l monitorizeze în timp ce el trimite e-mail-ul. Aceasta este o pierdere de timp pentru profesor, ținând cont de faptul că trebuie să treacă pe la fiecare elev și să urmărească aceeași procedură de fiecare dată. Acesta este de fapt rolul calculatoarelor în societatea de astăzi: pentru a efectua aceste sarcini repetitive. Abordarea noastră suplinește cu succes sarcina profesorului de evaluare a elevului, oferindu-i un instrument de evaluare automată, pe care îl vom detalia în secțiunea următoare.

IV Implementarea și testarea prototipului

Faza inițială a fost de proiectare a unui instrument de urmărire capabil să memoreze acțiunile unui utilizator de calculator și să fie, în același timp conștient de context [16]. Am dezvoltat un instrument capabil să analizeze acțiunile pe care un utilizator le efectuează pe un calculator cunoscând aplicația curentă și datele de intrare care sunt introduse în aplicație cu ajutorul mouse-ului și al tastaturii. Inițial am dezvoltat acest instrument folosind metode tradiționale, punând logica în codul sursă. Încercând să acoperim mai multe acțiuni și mai multe aplicații codul sursă a început să crească. De asemenea, aplicația a devenit mai lentă datorită faptului că numărul condițiilor a crescut exponențial. Având dovada că conceptului este funcțional și datorită faptului că rezultatele inițiale erau promițătoare, dar performanța se degrada, am început să luăm în considerare alte abordări pentru inteligența artificială a sistemului. Am decis că un sistem de raționament bazat pe o ontologie ar fi mai potrivit pentru nevoile noastre.

Ordinarily for this use-case you would need to give the learner a computer and have a teacher monitor him while he sends the e-mail. This is a waste of time for the teacher, taking into account the fact that he has to pass by every learner's computer and observe the same procedure over and over again. This is actually the role of computers in today's society: to perform such repetitive tasks. Our approach successfully makes the teacher's task of assessing the learner easier by providing him with a live assessment tool, which we detail in the next section.

IV PROTOTYPE IMPLEMENTATION AND USE CASE TESTING

The initial phase was the design of a tracking tool able to memorize the actions of a computer user and be at the same time context-aware [16]. We developed a tool able to analyse the actual actions that a user is performing on a computer with the use of knowledge of the currently focused application and the input that he fed into the application – mouse clicks and scrolls or keyboard's key presses. As a proof of concept we developed this tool using traditional methods, by putting the logic inside the code. As we tried to cover more and more actions and applications the code started to grow. It also became slower due to the fact that the number of “*if-else*” sequences was exponentially increasing. As we had the proof of concept up and running and the initial results were positive but the performance was falling, we started to take into account other approaches for the artificial intelligence of our system. We decided that an ontology-based reasoning system would be more suitable for our needs.

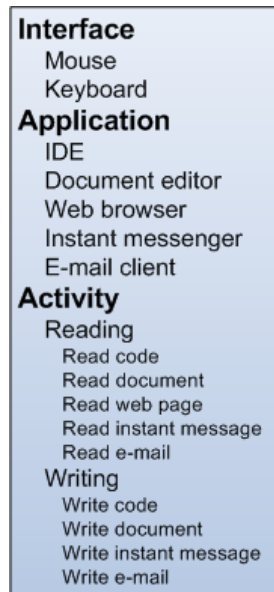


Figura 1: Categoriile principale ale ontologiei.
Figure 1: Ontology's Main Categories.

Motivul pentru care am decis să utilizăm o ontologie este că ea facilitează extinderea programul, fără a mai fi nevoie de o recompilare codului sursă. Astfel descrierea de noi aplicații și a acțiunilor se face în ontologie, salvând ontologia și punând noile fișierele RDF/RDFS generate în directorul programului acesta va ști automat cum să se evalueze noile acțiuni. Motivul pentru care acest lucru se întâmplă este că logica programului nu mai este în programul propriu-zis, ci în interiorul ontologiei. Ontologia, pe care am denumit-o "CONTEXT" modelează interacțiunea dintre om și calculator [2].

Taxonomia categoriilor de bază din ontologia noastră este prezentată în figura 1. Aceste categorii sunt disjuncte reciproc și acoperă întregul spațiu de obiecte.

Ontologia [7] are trei ramuri principale:

- ramura de interfețe - care mapează evenimentele de intrare generate de către dispozitivele de intrare (mouse-ul și tastatura); de exemplu, am mapat 522 și 256 întrerupe care corespund învârtirii roțiței mouse-ului și evenimentul apăsării unei taste în Windows.

The reason why we decided to use an ontology is that it will enable us to extend the program very easily, without having to recompile any piece of code. We just describe the new applications and actions in the ontology and put the newly generated RDF/RDFS files in the program's directory and the program will automatically know how to handle the newly added features. The reason why this happens is that the logic of the program does not reside in the program itself any more, but rather inside the ontology. The ontology, which we named "CONTEXT" models the human-computer interaction [2].

The taxonomy of the most basic categories of particulars in our ontology is depicted in Figure 1. They are mutually disjoint and cover the whole domain of particulars.

The ontology [7] has three main branches:

- The interface branch – which maps the input events raised by the input devices (mouse & keyboard); for example we map the 522 and 256 interrupts which correspond to mouse scroll and key down events in Windows.

- ramura aplicații - care creează o ierarhie de aplicații în funcție de tipul acesteia (ex. editor de documente - Microsoft Word, client e-mail - Microsoft Outlook).
 - ramura de activități - este ramura principală a ontologie. Aceasta mapează activitățile utilizatorului atât pe baza aplicației active cât și a interfeței utilizate la un moment dat, pentru o anumită sarcină. De exemplu, în cazul în care aplicația Microsoft Outlook este activă și evenimentului 522 al tastaturii - de apăsare a unei taste are loc, atunci putem crea triplete asociate acestei stări pe care să le adăugăm la graful ontologiei noastre, mapând astfel aplicația - Microsoft Outlook cu evenimentul de apăsare a unei taste. Graful rezultat oferă o imagine a activităților elevului de-a lungul timpului și permite interogări interesante care urmează să fie efectuate pe acesta, folosind SPARQL pentru a oferi diverse unghiuri de vedere. Pentru cazul nostru particular, graful rezultat conține doar un singur nod cu valoarea: "Utilizatorul scrie un e-mail în Microsoft Outlook".
- The application branch – which creates a hierarchy of applications based on the application type(ex. document editor – Microsoft Word, e-mail client – Microsoft Outlook).
 - The activity branch – is the main branch in the ontology. It maps the activities of a user based on both the active application and interface currently being used for a certain task. For instance, if the Microsoft Outlook application is active and the 522 mouse scroll event occurs then we can create associated triples to add to our graph to log the application - Microsoft Outlook and the event - mouse scroll. The resulting graph provides a picture of the user’s activities over time and allows interesting queries to be performed on it using SPARQL to provide views from numerous angles. For our special case the resulting graph contains just one node with the value: “User writes an e-mail in Microsoft Outlook”.

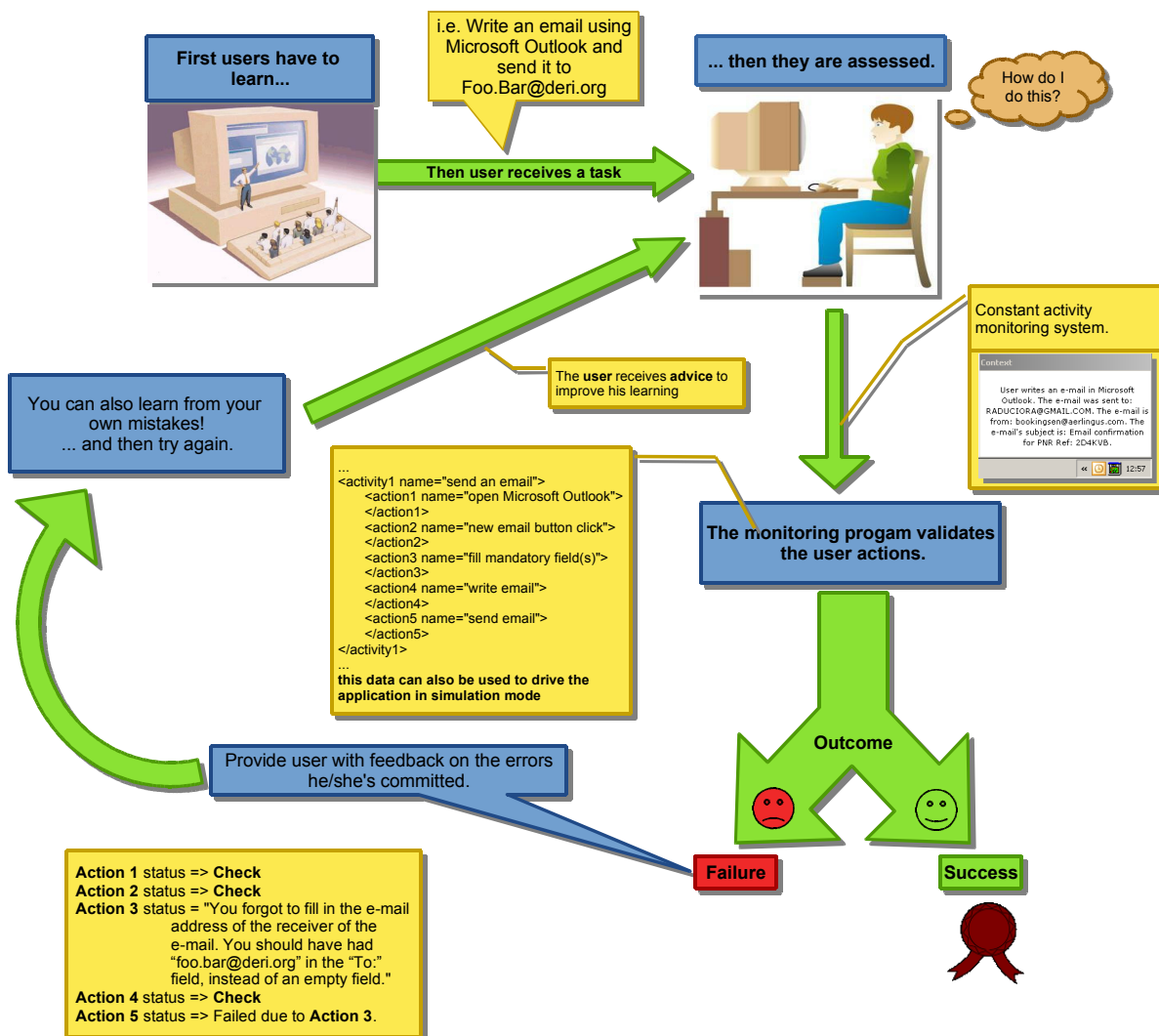


Figura 2: Evaluarea automată cu feedback.
 Figure 2: Live Assessment with Learning Help and Simulation.

Ontologia a fost proiectată în Protégé și exportată în format RDF/RDFS [1] [3], considerat a fi cel mai potrivit pentru acest caz. Pentru achiziția datelor despre acțiunile elevului în scopul de a le compara cu acțiunile din ontologie, am folosit trei tipuri de interfețe.

Prima interfață folosită a fost interfața de automatizare OLE [18]. Această interfață se bazează pe Component Object Model (COM). COM [12] este o arhitectura software standard pe baza de interfețe care este proiectată pentru a separa codul în obiecte cu conținut închis, sau componente. Fiecare componentă expune un set de interfețe de comunicare prin care se poate manipula

The ontology was designed in Protégé and exported it in RDF/RDFS [1][3] format, which we considered to be the most appropriate for this case. In order to receive the learner's feedback in order to compare the actions in the ontology with the actual ones performed by him, we used three types of interfaces.

The first interface we used was the OLE automation interface [18]. This interface is based on the Component Object Model (COM). COM [12] is a standard software architecture based on interfaces that is designed to separate code into self-contained objects, or components. Each component exposes a set of interfaces through which all

respectiva componentă. Scopul principal al interfeței de automatizare OLE este de automatizare a aplicațiilor. Noi nu ne-am concentrat pe această latură a interfeței, ci mai degrabă pe informațiile pe care aceasta le oferă despre aplicațiile cu care aceasta ne conectează. Aceste informații pot fi de la autorul unui document până la subiectul unui e-mail.

O a doua interfață folosită a fost interfața de „hooking” [13]. Un „hook”(cârlig) este un punct în mecanismul de tratarea a mesajelor de sistem, unde o aplicație poate instala o subrutină pentru a monitoriza traficul de mesaje din sistem și să proceseze anumite tipuri de mesaje. Sistemul suportă mai multe tipuri de cârlige, dar interesul nostru a fost limitat în monitorizarea mouse-ului și al tastaturii. Această interfață a permis accesul direct la dispozitivelor de intrare. Acest lucru a fost util într-o anumită măsură. Astfel am putut afla ce, unde și când utilizatorul a făcut clic sau a tastat ceva, dar nu ne-ar da informații legat de elementul din interfață cu care utilizatorul a interacționat.

Pentru a putea spune care element al interfeței cu utilizatorul a fost acționat avem nevoie de ajutorul celui de al treilea API numit Active Accessibility [14]. Active Accessibility este menită să ajute persoanele cu dizabilități. Aici ajută aplicația noastră să interacționeze cu elementele din interfața cu utilizatorul al altor aplicații și cu sistemul de operare. Astfel această interfață ne oferă informații legate de elementul din interfața cu utilizatorul acționat de un utilizator, la un moment dat.

Cu ajutorul acestor trei interfețe suntem capabili de a captura toate acțiunile pe care un elev le efectuează pe un computer. Am extins instrumentul cu o funcționalitate de arhivare, pentru a păstra o istorie a acțiunilor elevului. Pe baza acestui instrument de urmărire de nivel inferior am construit un instrument de evaluare, care are ca date de intrare un document XML. Documentul XML conține descrierea unei sarcini pe care un cursant trebuie să o rezolve. Un exemplu de

communication to the component is handled. The OLE automation interface’s primary goal is automating applications. We don’t focus on this part of the interface, but rather on the information that it can provide about the applications it connects us to. This information goes from the Word document’s author to the subject or the content of an e-mail.

A second interface we used was the hooking interface [13]. A hook is a point in the system message handling mechanism where an application can install a subroutine to monitor message traffic in the system and process certain types of messages. The system supports many types of hooks, but our interest was limited in hooking the mouse and the keyboard. This interface gave us access to the raw input of the input devices; this was useful at some extent - meaning that we could find out what, where and when the user clicked something, but it would not give us the information which UI element was activated.

For being able to tell which UI element was activated we need the aid of a third API called Active Accessibility [14]. The Active Accessibility is meant to help accessibility aids, here our application, interact with UI elements of other applications and the operating system. So this interface allows us to tell which UI element a user activated at a given time.

With the aid of these three interfaces we are able to capture all the actions a learner performs on a computer. We extended the tool with a logging functionality, so we could keep a history of the learner’s actions. On top of this low-level tracking tool we buildt an assessment tool, which has as input an XML document. The XML document contains the description of a task that a learner has to solve. A sample of such a XML document can be seen in Figure 2. Our program sits in background and monitors the learner’s actions. When the learner closes the application, A SPARQL query is being generated with the collected information from

astfel de document XML se poate observa în figura 2. Programul nostru stă în fundal și monitorizează acțiunile celui evaluat. Când elevul închide aplicația, o interogare SPARQL este generată cu informațiile colectate de la acțiunile sale. Cu aceasta se interoghează ontologia. Dacă evaluatul a executat corect sarcina, adică dacă el a efectuat toate acțiunile necesare în ordinea corectă, ontologia ne spune că elevul a terminat cu succes testul. Dacă acesta a greșit, rezultatul interogării va fi feedback-ul pentru erorile comise.

Experimentele în evaluarea elevului au arătat rezultate pozitive, care fac din acest instrument unul aplicabil pe scară largă. Am creat o serie de scenarii din lumea reală care să testeze prototipul nostru. Dacă luăm analizăm cazul trimiterii unui e-mail, acesta ilustrează în sistemul nostru fluxul general al evaluării automate. Deoarece abordarea noastră este conștientă de context, sistemului de monitorizare este suficient de inteligent pentru a nu suprasolicita sistemul cu prelucrări inutile de date, prin filtrarea evenimentelor irelevante. În acest fel, monitorizarea evenimentelor ignoră acele evenimente care nu sunt relevante pentru sarcina curentă evaluată. De exemplu, dacă un utilizator schimbă contextul (trece la o altă aplicație), în timpul testării și începe să facă altceva, sistemul de monitorizare nu mai procesează acțiunile sale, dar va relua monitorizare atunci când el comută înapoi la aplicația supusă evaluării [17].

V Concluzii și dezvoltări ulterioare

Am descris un sistem care evaluează în mod automat un cursant și îi oferă feedback, fără nici o interacțiune umană. Sistemele anterioare tratau problema de verificare a cunoștințelor prin intermediul unor chestionare cu variante multiple sau erau de tipul întrebare și răspuns. Noi credem că un instrument de evaluare practică este mult mai atrăgător și mai educativ decât evaluarea cunoștințelor teoretice. Ușurința de extindere

learner's actions. With this query we interrogate the ontology. If he executed the tasks correctly – meaning if he completed all the necessary steps in the required order, the ontology tells the learner that he successfully completed the test. If he failed to do so, the query result will be feedback on the errors he committed.

Experiments in assessing the learner have shown positive results which make this tool widely applicable. We have created a set of use cases based on real world scenarios that is able to test our prototype. If we take the use case of sending an e-mail, it illustrates the overall workflow of the automated assessment in our system. As our approach is context aware the monitoring system is smart enough not to overload the system with unnecessary processing, by filtering the irrelevant events. In this way, the tracking ignores the events that are not relevant to the current assessed task, for example if a user switched context (switched to a different application), while he was in the middle of test and started doing something else, the system no longer processes his actions, but will resume monitoring as soon as he switches back to the given task [17].

V CONCLUSIONS AND FUTURE WORK

We have described a system that automatically assesses a learner and provides him with feedback, without any human interaction. Previous systems treat the problem of assessment using multiple choice questionnaires or were question-answering based. We believe that a live assessment tool is much more engaging and more educative than theoretical knowledge assessment. The easiness of extending the system either with new tests – based on XML input or to cover new applications – based on the ontology, makes it very scalable.

In future we want to extend the system to be able to assess multiple tasks at the same time. Additionally, we intend to implement a simulation module which would show a

a sistemului, fie cu noi evaluări - bazat pe XML de intrare sau pentru a acoperi noi aplicații - bazate pe extinderea ontologiei, îl face foarte scalabil.

În viitor dorim să extindem sistemul să fie în măsură să evalueze mai multe sarcini în același timp. În plus, ne-am propus să punem în funcțiune un modul de simulare care să arate elevului modul corect de îndeplinire a unei sarcini date.

VI Mulțumiri

Dorim să mulțumim lui SP Williams pentru feedback-ul lui valoros. Această lucrare a fost susținută prin Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007-2013 Axa prioritară 1: "Educația și formarea profesională în sprijinul creșterii economice și dezvoltării societății bazate pe cunoaștere", domeniul major de intervenție 1.5: "Programe doctorale și post-doctorale de studii pentru sprijinire a cercetării". Titlul proiectului este "de armonizare a valențelor academice românești cu cele ale Comunității Europene". Codul de contract: POSDRU/CPP107/DMI1.5/S/76851, beneficiar Universitatea Lucian Blaga din Sibiu.

BIBLIOGRAFIE

- [1] Beckett, D., *RDF/XML Syntax Specification (Revised)*, W3C Recommendation, 10 februarie 2004. <http://www.w3.org/TR/rdf-syntax-grammar/>
- [2] Berners-Lee, T. Hendler, J. Lassila, O., *The Semantic Web*, Scientific American, mai 2001.
- [3] Brickley, D. and Guha, R.V., *RDF Vocabulary Description Language 1.0: RDF Schema*, W3C Recommendation, 2004. <http://www.w3.org/TR/rdf-schema/>
- [4] DTI 2006 - *Beyond eLearning: practical insights from the USA*.
- [5] Extensible Markup Language (XML). <http://www.w3.org/XML/>
- [6] FREMA - e-learning Framework

learner the necessary steps that need to be performed in order to fulfil a given task.

VI ACKNOWLEDGEMENTS

We gratefully acknowledge S.P. Williams for his valuable feedback. This work was supported by Sectoral Operational Programme Human Resources Development 2007-2013 Priority Axis 1: "Education and training in support for growth and development of a knowledge based society", key area of intervention 1.5: "Doctoral and post-doctoral studies programmes for research support". The project title is "Harmonization of the Romanian academic valences of the European Community". Contract code: POSDRU/CPP107/DMI1.5/S/76851 and the beneficiary is Lucian Blaga University of Sibiu.

BIBLIOGRAPHY

- [1] Beckett, D., *RDF/XML Syntax Specification (Revised)*, W3C Recommendation, 10 February 2004. <http://www.w3.org/TR/rdf-syntax-grammar/>
- [2] Berners-Lee, T. Hendler, J. Lassila, O., *The Semantic Web*, Scientific American, May 2001.
- [3] Brickley, D. and Guha, R.V., *RDF Vocabulary Description Language 1.0: RDF Schema*, W3C Recommendation, 2004. <http://www.w3.org/TR/rdf-schema/>
- [4] DTI 2006 - *Beyond eLearning: practical insights from the USA*.
- [5] Extensible Markup Language (XML). <http://www.w3.org/XML/>
- [6] FREMA - e-learning Framework Reference Model for Assessment, http://www.jisc.ac.uk/index.cfm?name=elfref_frema&src=alp
- [7] Gomez-Perez, A. Corcho, O. Fernandez-Lopez, M., *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*, Springer, 2004.
- [8] Livingstone, D. et al., *Automated System*

- Reference Model for Assessment,
http://www.jisc.ac.uk/index.cfm?name=elfref_frema&src=alp
- [7] Gomez-Perez, A. Corcho, O. Fernandez-Lopez, M., *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*, Springer, 2004.
- [8] Livingstone, D. et al., *Automated System for the Assessment of Programming*, 2004.
- [9] Livingstone, D. et al., *JISC e-Learning Framework Assessment Demonstrator*, 2005.
- [10] Marshall, A. et al., *Tracking and Reporting in e-Learning Contexts*, 2006.
- [11] Moran, T.P., *Context in Design*. Human-Computer Interaction, 1994. (Special Issue on context in design.).
- [12] MSDN, COM - The Common Object Model.
[http://msdn2.microsoft.com/en-us/library/ms694363\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/ms694363(VS.85).aspx)
- [13] MSDN, Hooks.
[http://msdn2.microsoft.com/en-us/library/ms632589\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/ms632589(VS.85).aspx)
- [14] MSDN, How Active Accessibility Works.
[http://msdn2.microsoft.com/en-us/library/ms697335\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/ms697335(VS.85).aspx)
- [15] Prud'hommeaux, E. Seaborne, A., *SPARQL Query Language for RDF*, W3C Re-commendation, 2008.
<http://www.w3.org/TR/rdf-sparql-query/>
- [16] Roman, G.-C., Julien, C. and Payton, J., *A Formal Treatment of Context Awareness*. in *7th International Conference on Fundamental Approaches to Software Engineering (FASE'04)*, 2004, Barcelona, Spain.
- [17] Schmidt, A., *Potentials and Challenges of Context-Awareness for Learning Solutions*, in *Proceedings of the 13th Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems*, 2005.
- [18] Turner, L., *Automating Microsoft Office 97 and Microsoft Office 2000*
[http://msdn2.microsoft.com/en-us/library/aa155776\(office.10\).aspx](http://msdn2.microsoft.com/en-us/library/aa155776(office.10).aspx)
- for the Assessment of Programming*, 2004.
- [9] Livingstone, D. et al., *JISC e-Learning Framework Assessment Demonstrator*, 2005.
- [10] Marshall, A. et al., *Tracking and Reporting in e-Learning Contexts*, 2006.
- [11] Moran, T.P., *Context in Design*. Human-Computer Interaction, 1994. (Special Issue on context in design.).
- [12] MSDN, COM - The Common Object Model.
[http://msdn2.microsoft.com/en-us/library/ms694363\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/ms694363(VS.85).aspx)
- [13] MSDN, Hooks.
[http://msdn2.microsoft.com/en-us/library/ms632589\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/ms632589(VS.85).aspx)
- [14] MSDN, How Active Accessibility Works.
[http://msdn2.microsoft.com/en-us/library/ms697335\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/ms697335(VS.85).aspx)
- [15] Prud'hommeaux, E. Seaborne, A., *SPARQL Query Language for RDF*, W3C Re-commendation, 2008.
<http://www.w3.org/TR/rdf-sparql-query/>
- [16] Roman, G.-C., Julien, C. and Payton, J., *A Formal Treatment of Context Awareness*. in *7th International Conference on Fundamental Approaches to Software Engineering (FASE'04)*, 2004, Barcelona, Spain.
- [17] Schmidt, A., *Potentials and Challenges of Context-Awareness for Learning Solutions*, in *Proceedings of the 13th Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems*, 2005.
- [18] Turner, L., *Automating Microsoft Office 97 and Microsoft Office 2000*
[http://msdn2.microsoft.com/en-us/library/aa155776\(office.10\).aspx](http://msdn2.microsoft.com/en-us/library/aa155776(office.10).aspx)