Studio: Ontology-Based Educational Self-Assessment

Christian Weber
Corvinno Technology
Transfer Center
Budapest, Hungary
cweber@corvinno.com

Réka Vas
Corvinus University
of Budapest
Budapest, Hungary
reka.vas@uni-corvinus.hu

ABSTRACT

Students, through all stages of education, grasp new knowledge in the context of knowledge memorized all through their previous education. To self-predict personal proficiency in education, self-assessment acts as an important learning feedback. The in-house developed Studio suit for educational self-assessment enables to model the educational domain as an ontology-based knowledge structure, connecting assessment questions and learning material to each element in the ontology. Self-assessment tests are then created by utilizing a sub-ontology, which frames a tailored testing environment fitting to the targeted educational field. In this paper we give an overview of how the educational data is modeled as a domain ontology and present the concepts of different relations used in the Studio system. We will deduct how the presented self-assessment makes use of the knowledge structure for online testing and how it adapts the test to the performance of the student. Further we highlight where potentials are for the next stages of development.

Keywords

Education, adaptive test, self-assessment, educational ontology

1. INTRODUCTION

Students exploring new fields of education are always confronted with questions regarding their individual progress: how much do they know after iterations of learning, in which directions should they progress to fill the field most effectively, how to grasp the outline and details of the field and how much of their time should they invest in learning? Especially in higher education, where learning becomes a self-moderated, personalized process, students are in need of continuous self-assessment to capture their current state of proficiency. At the same time, the unframed, informal self-prediction of students regarding their personal skills is often substantive and systematically flawed [1]. Here a systematic and objective solution for self-assessment is substantial to prevent a wrong or biased self-evaluation and to support the self-prediction of the personal proficiency.

Following Jonassen, knowledge in education could be split into nine types across three categories to capture the human's cognitive behavior. In his discussion, eight out of nine knowledge types underline that knowledge in the scope of learning is interrelated and strongly associated with previous experiences [2]. As such, a supporting solution for self-assessment should grasp and formalize the knowledge to assess in the context of related knowledge.

The Studio suit for educational self-assessment, presented in this paper, provides here a software solution for testing the personal proficiency in the context of related knowledge. It enables to model areas of education as a substantial source for assessment and narrows the gap between a potentially flawed self-prediction and the real proficiency, by offering an objective and adaptive online knowledge-test. To follow the natural learning process and enable an easy extension, the software embeds the assessed knowledge into a network of contextual knowledge, which enables to adapt the assessment to the responses of the students.

This paper will give an overview of the Studio educational domain ontology and the aspects of the system supporting personalized self-assessment. Further it will highlight potentials for data mining on the gathered educational data with an outlook on the next stages of evaluation.

2. THE STUDIO APPROACH FOR SELF-ASSESSMENT

The basic concept of Studio is to model the focused education as an interrelated knowledge structure, which divides the education into sub-areas and knowledge items to know. The managed structure formalizes the relation between knowledge areas as a learning context and models the requirements to master specific parts of the education. This structure is used to create and support knowledge tests for students. Through this combination of assessment and knowledge structure, the student gains the freedom to explore not only single knowledge items but the education in the context of related knowledge areas, while the embedded requirements are used to map the modeled knowledge against the expected educational outcome.

The assessment-system is designed to be accompanied by phases of learning within the system, where the student gets access to learning material, based on and supported by the test feedback. This combined approach offers a unique self-assessment to the students, where the backing knowledge context is used to adapt the assessment in dependency of the test performance of the student.

Before any regular examination students may use Studio to assess their knowledge on their own. It is the tutor's responsibility to set the course of self-assessment test in Studio system by selecting knowledge areas and sub-knowledge areas which are relevant for the target education from the domain ontology. Then the frame will be automatically completed with elements from the ontology which detail the selected knowledge areas and are modeled as required for this part of the education.

As the system stores assessment questions for each knowledge element, Studio will then automatically prepare an assessment test, based on the defined selection and the domain ontology. The resulting knowledge-test is then accessible as a self-assessment test for the student, who explores the backed knowledge structure, which pictures the expected learning outcome, in cycles of testing, reflection and learning. The process of test definition and assessment is shown in Figure 1, while the result preparation for reflection and learning is discussed in section 2.5.

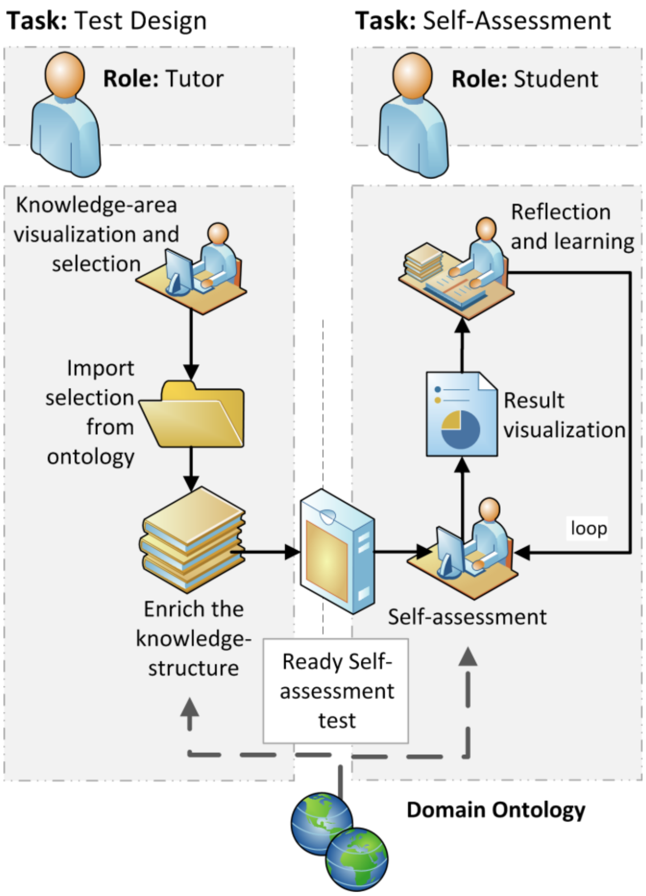

Figure 1: The overall design, assess and reflection cyle of the system.

2.1 The Educational Domain Ontology

The Studio system is based on a predesigned educational ontology, explained in detail by Vas in [3]. Domain ontology is a frequently used term in the field of semantic technologies and underlines the storage and conceptualization of domain knowledge and is often used in a number of projects and solutions [4][5][6] and could address a variety of domains with different characteristics in their creation, structure and granularity, depending on the aim and the modeling person [7]. A specialization in terms of the field is the educational domain ontology which is a domain ontology adapted to the area and concepts of education. They could target to model different aspects of education as the curriculum or aspects relevant for the task of learning and course creation [8][9][10] or describe the design, use and retrieval of learning materials till creating courses [11], as well as directly the learner within the education [12].

Within the area of educational ontologies, domain ontologies tend to model too specific details of the education, in an attempt to model the specific field as complete as possible. This enables a comprehensive view on the field but it comes at the cost of generality, with the potential to be inflexible to handle changes. Other concepts model the education across different ontologies, matching concepts like the learner, the education and the course description, introducing a broad horizon but with additional overhead to combine modelled insights and reason on new instances.

The appeal of the Studio educational ontology is the size and focus of the main classes and their relationships between each other. The knowledge to learn is the main connecting concept in the core of education. It enables a great flexibility to be resourceful for different education related questions. An example is here the business process management extension PROKEX, which maps process requirements against knowledge areas to create assessment test, reflecting the requirements of attached processes [13].

An important factor in learning is the distance between the expectation of the tutor and the learning performance of the student. Here a short cycle of repeated assessment and learning is a major factor for a better personal learning performance [14]. This aspect directly benefits from the focused concentration on knowledge-areas as the main exchange concept between students and tutors. As even further the close connections between learners and educators via direct tutoring is one major enabler for computer aided systems [15], each step towards a more direct interaction through focused concepts is an additional supporter.

The class structure fuses the idea of interrelated knowledge with a model of the basic types of educational concepts, involved in situations of individual learning. Figure 2 visualizes the class concepts as knowledge elements, together with the relation types, used to model the dependencies between different aspects of knowledge and learning within the educational ontology.

The Knowledge Area is the super-class and core-concept of the ontology. The ontology defines two qualities of main relations between knowledge areas: Knowledge areas could be a sub-knowledge area of other knowledge areas with the “has_sub-knowledge_area” relation or be required for another knowledge area with the “requires_knowledge_of” relation. A knowledge area may have multiple connected knowledge areas, linked as a requirement or sub-area. The “requires_knowledge_of” relation defines that a node is required to complete the knowledge of a parent knowledge area. This strict concept models a requirement dependency between fields of knowledge in education and yields the potential to assess perquisites of learning, analog to the basic idea of perquisites within knowledge spaces, developed by Falmagne [16].

Education is a structured process which splits the knowledge to learn into different sub-aspects of learning. Knowledge areas in the ontology are extended by an additional sub-layer of knowledge elements in order to effectively support educational and testing requirements. Figure 2 visualizes the sub-elements and their relations. By splitting the assessed knowledge into sub-concepts, the coherence and correlation of self-assessment questions could be expressed more efficiently and with the potential of a more detailed educational feedback.

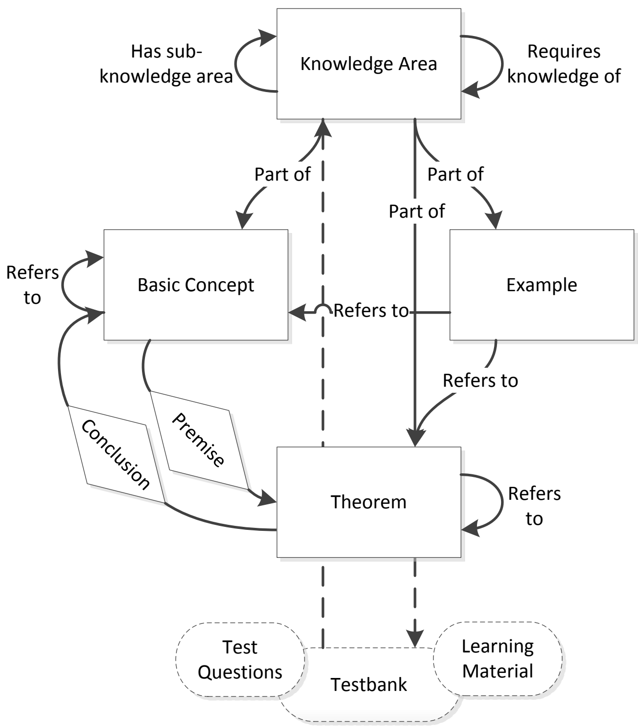

Figure 2: Model of the educational ontology.

Theorems express in a condensed and structured way the fundamental insights within knowledge areas. They fuse and explain the basic concepts of the depicted knowledge and set them in relation to the environment of learning with examples. Multiple theorems could be "part_of" a knowledge area. Each theorem may define multiple Basic Concepts as a "premise" or "conclusion", to structure how the parts of the knowledge area are related. Examples enhance this parts as a strong anchor for self-assessment questions and "refer_to" the theorems and basic concepts as a "part_of" one or more knowledge areas.

2.2 The Testbank

In order to connect the task of self-assessment with the model of the educational domain, the system integrates a repository of assessment questions. Each question addresses one element of the overall knowledge and is directly associated with one knowledge area or knowledge element instance within the ontology. The domain ontology provides here the structure for the online self-assessment while the repository of questions supplements the areas as a test bank. The target of the self-assessment is to continuously improve the personal knowledge within the assessed educational areas, by providing feedback on the performance after each phase of testing. To do so, the Studio system includes Learning Material connected to the test bank and the knowledge areas, analog to the test questions. The learning material is organized into sections as a structured text with mixed media, as pictures and videos, and is based on a wiki-engine to maintain the content, including external links.

2.3 Creating and Maintaining Tests

The creation and continues maintenance of the domain ontology is a task of ontology engineering. The ontology engineer (the ontologist), creates, uses and evaluates the ontology [17], with a strong focus on maintaining the structure and content. Within Studio, this process is guided and supported by a specialized administration workflow and splits in three consecutive task areas, in line with decreasing access rights:

- **Ontology engineering (instance level):** The creation and linking of instances of the existing knowledge-area classes into the overall domain ontology.
- **Test definition:** Knowledge areas, which are relevant to a target self-assessment test, are selected and grouped into specialized containers called Concept Groups (CG). These concept groups are organized into a tree of groups, in line with the target of the assessment. The final tree in this regards captures a sub-ontology. Concept groups are internally organized based on the overall ontology and include all relations between knowledge elements, as defined within the domain ontology.
- **Question and learning material creation:** Questions and learning materials alike are directly connected to single knowledge areas within the designed test frame and get imported, if already existing, from the domain ontology. More questions and learning materials are defined now, in line with the additional need of the targeted education and are available for future tests.

The pre-developed structure of classes and relations is fixed as the central and integral design of the system. A view of the system interface for administration is provided in Figure 3. The left area shows the visualization of the current ontology section in revision and the right area shows the question overview with editing options. Tabs give access to additional editing views, including the learning material management and interfaces to modify relations between nodes and node descriptions.

2.4 Adaptive Self-Assessment

To prepare an online self-assessment test, the system has to load the relevant educational areas from the domain ontology and extract the questions and relations of the filtered knowledge areas.

The internal test algorithm makes use of two assumptions:

- **Knowledge-area ordering:** As the main knowledge areas are connected through "requires_knowledge_of" and "part_of" relations, every path, starting with the start-element, will develop on average from general concepts to detailed concepts - given that the concept groups in the test definition are also selected and ordered to lead from general to more detailed groups.
- **Knowledge evaluation dependency:** If a person, taking the test, fails on general concepts he or she will potentially also fail on more detailed concepts. Further, if a high number of detailed concepts are failed, the parent knowledge isn't sufficiently covered and will be derived as failed, too.

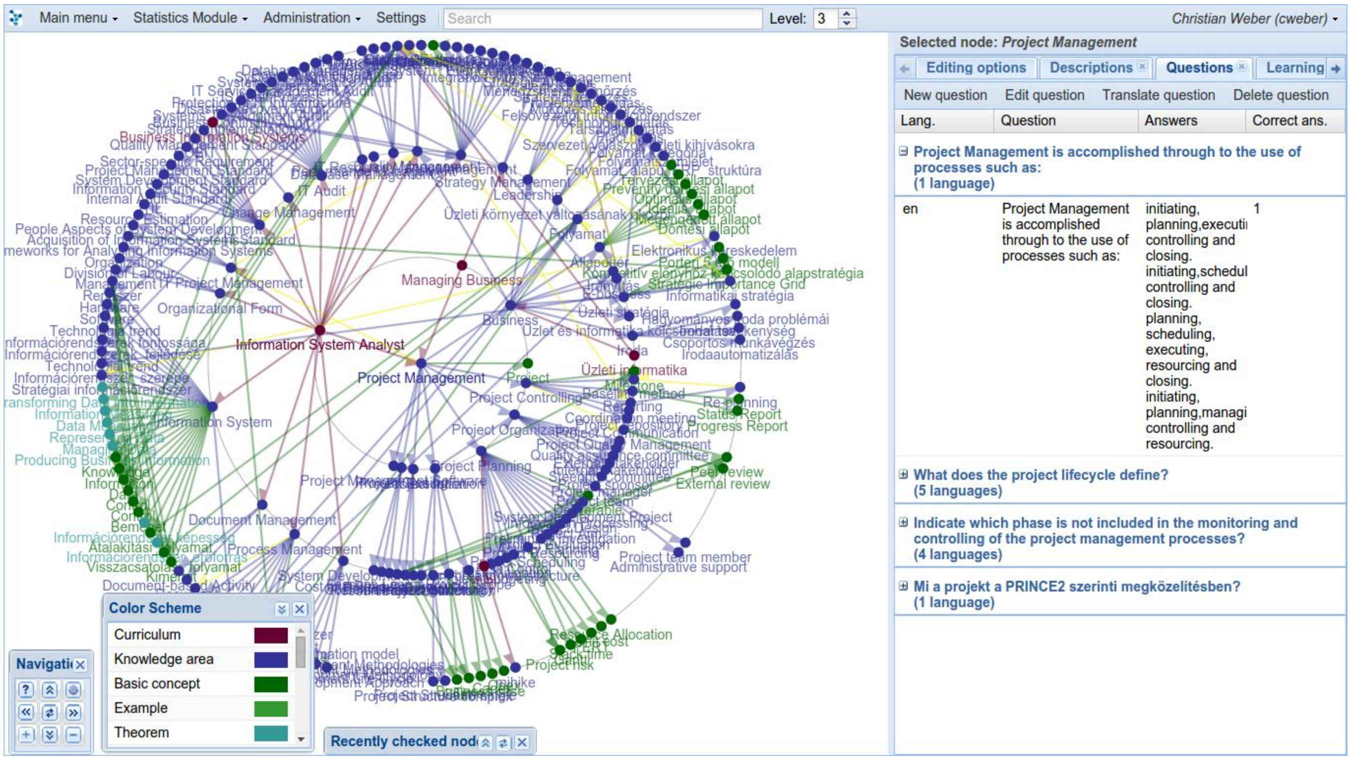

Figure 3: The main ontology maintenance and administration interface, showing a part of the domain ontology.

The filtering is done based on the selection of a tutor, acting as an expert for the target educational area. The tutor chooses related areas, which are then created as a Test Definition, containing Concept Groups, as described in section 2.3. The system then uses the test definition as a filtering list to extract knowledge areas. After the extraction, the structure is cached as a directed graph, while the top element of the initial concept group is set as a start element. Beginning with the start-element, the test will move then through the graph, while administering the questions connected to knowledge areas and knowledge elements.

The loading of knowledge-elements follows three steps:

1. Each type of relation between two knowledge-elements implements a direction for the connection. Assuming the system loads all relations, starting with the start-element and ending on a knowledge-element, this creates a two level structure where the start-node is a parent-element and all related, loaded elements are child-elements, as seen below in Figure 4.
2. The loading algorithm then selects one child-element and assumes it as a start-element and repeat the loading process of knowledge-elements.
3. When no knowledge-elements for a parent-element could be loaded, the sub-process stops. When all sub-processes have stopped, the knowledge structure is fully covered.

The test algorithm will now activate the child knowledge areas of the start element and select the first knowledge area to the left and draw a random question from the selected knowledge area. If the learner fails the question, the algorithm will mark the element as failed and selects the next knowledge area from the same level. If the learner's answer is correct, the system will activate the child elements of the current node and draw a random question from the first left child.

Based on the tree shaped knowledge structure, the assessment now follows these steps to run the self-assessment, supported by the extracted knowledge structure:

1. Starting from the start-element, the test algorithm will activate the child knowledge-areas of the start element.
2. The algorithm now selects the first child-knowledge area and draws a random question out of the pool of available questions for this specific knowledge-element from the test bank.
3. If the learner fails the question, the algorithm will mark the element as failed and select the next knowledge area from the same level. If the learner's answer is correct, the system will activate the child elements of the current node and trigger the process for each child-element.

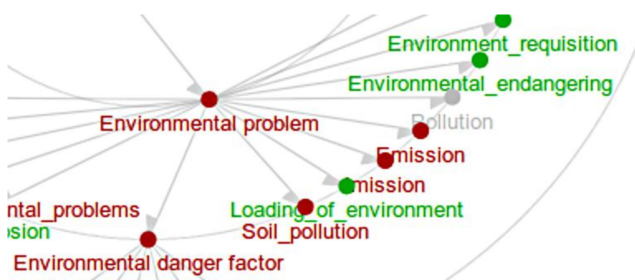

Figure 4: Excerpt from the sub-ontology visualization, with the visible parent-child relationship, as used in the data-loading for preparing the self-assessment.

An example question is shown below in Figure 5. Further following the testing algorithm, the system dives down within the domain ontology and triggers questions depending on the learner's answers and the extracted model of the relevant education. In this regards the Studio system adapts the test on the fly to the performance of the learner. Correlating to the idea of adaptation, the learner will later gain access to learning material for each mastered knowledge area. As the learner continues to use the self assessment to evaluate the personal knowledge, he or she will thus explore different areas of the target education, following their individual pace of learning.

Figure 5: Test interface with a random drawn test question.

2.5 Test Feedback and Result Visualization

An important aspect of the system is the test feedback and evaluation interface. The educational feedback is one of the main enabler for the student to grasp the current state and extend of the personal education. The domain ontology models the structure and the dependencies of the educational domain, and the grouped test definition extracts the relevant knowledge for the target area or education. As such, the visualization of the ontology structure extracted for the test, together with the indication of correct and incorrect answers, represents a map of the knowledge of the learner.

Throughout each view onto the ontology, the system uses the same basic visualization, making use of the Sencha Ext JS JavaScript framework [18]. The visualization itself is a custom build, similar to the Ext JS graph function "Radar" and based on the idea of Ka-Ping, Fisher, Dhamija and Hearst [19]. All views are able to zoom in and out of the graph, move the current excerpt and offer a color code legend, explaining the meaning of the colored nodes. In comparison with state of the art, the interface offers no special grouping or additional visualization features like coding information into the size of nodes. Each interface offers an additional textual tree view to explore the knowledge-elements or concept groups in a hierarchical listing. This simple, straightforward approach for visualization correlates with the goal of a direct and easy to grasp feedback through interfaces which have a flat learning curve and enable to catch the functionality in a small amount of time.

While this simple visualization is sufficient for the reasonable amount of knowledge-elments within the result view, this alone is not suitable for the domain ontology administration interface, as seen in Figure 3. Here Studio realizes methodologies to filter and transform the data to visualize. To do so it makes use of two supporting mechanisms:

- The **maximum-level-selector** defines the maximum level the system extracts from the domain ontology for full screen visualization.
- In combination with the maximum level, the ontologist could select single elements within the domain ontology. This triggers an **on-demand re-extraction of the visualized data**, setting the selected knowledge-element as the centre element. The system then loads the connected nodes, based on their relations into the orientation circles till the maximum defined level is reached. More details about the transformation are in [19].

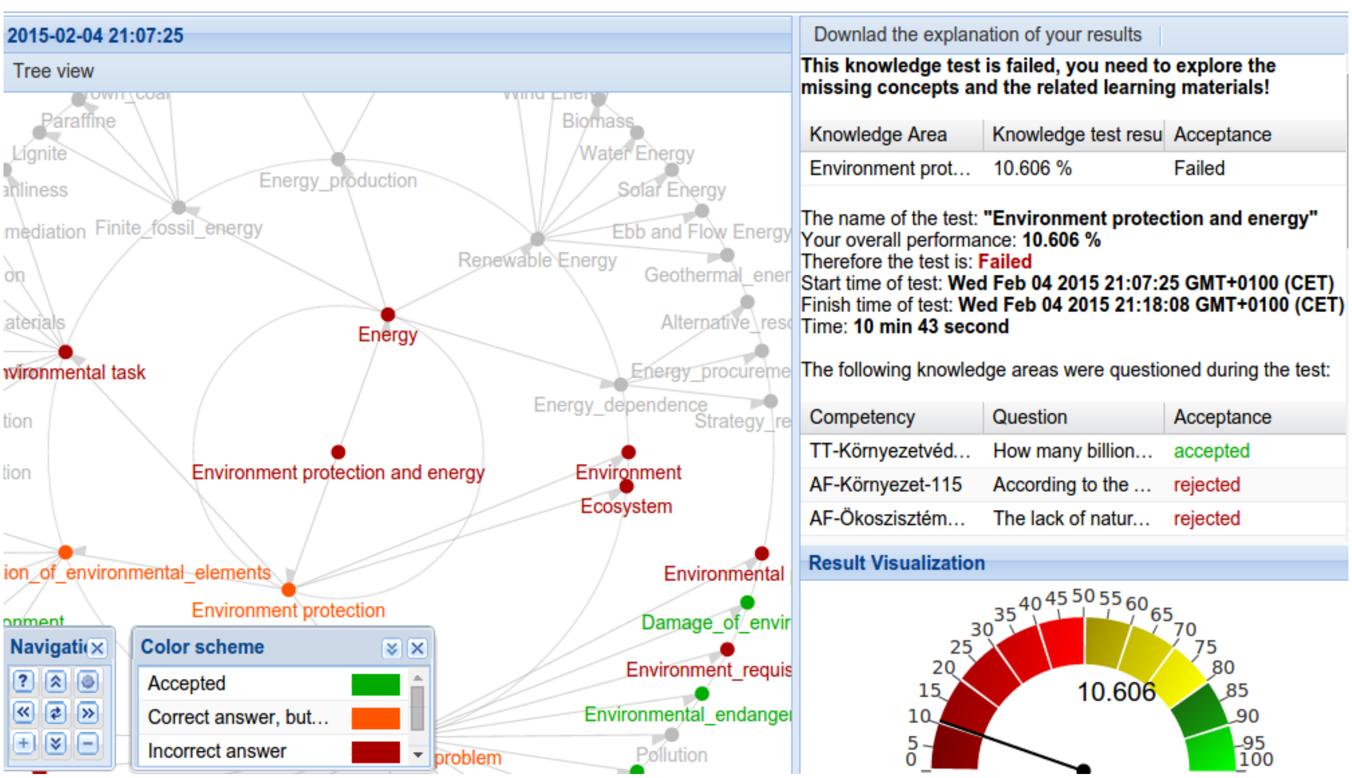

Figure 6: Result visualization as educational feedback for the learner.

Together, this selection and transformation mechanism enables the fluent navigation within the complete domain ontology structure, while re-using the same visualization interface.

Figure 6 shows the main view of the result interface. The left area shows the sub-ontology extracted for the test, while the colored nodes represent the answers to the administered questions. A red node visualizes wrong answers, while orange nodes are rejected nodes with correct answers but with an insufficient number of correctly answered child nodes, indicating a lack of the underlying knowledge. Green nodes represent accepted nodes with correct answers and a sufficient amount of correctly answered questions for child nodes. Grey nodes are not administered nodes, which were not yet reached by the learner, as higher order nodes had no adequate acceptance.

Even though the target of the system is not a strict evaluation in number, the evaluation of the percentage of solved and accepted knowledge elements helps the learner to track the personal progress and could additionally be saved as a report for further consultation. Besides providing an overview of the self-assessment result, the result interface gives access to the integrated learning material. For every passed node, the learner can now open the correlated material and intensify the knowledge for successful tested areas.

Retaking the test in cycles of testing and learning, while adapting the educational interaction, is the central concept of the Studio approach for self-assessment. As a consequence the system will not disclose the right answers to questions or learning material for not yet administered knowledge areas, to promote an individual reflection on the educational content outside of a flat memorization of content.

3. SYSTEM EVALUATION

The system has been used, extended and evaluated in a number of European and nationally funded research projects, including applications in business process management and innovation-transfer [20], medical education [21] and job market competency matching [22].

Currently the system is being evaluated based on a running study with 200 university students in the field of business informatics. The study will conclude on two current research streams which are improving the systems testing and analysis capability. The first direction looks into potentials for the integration of learning styles into adaptive learning systems to offer valuable advice and instructions to teachers and students [23]. Within the second direction the question is challenged on how to adapt the presented self-assessment further towards the performance of the students, based on extracting assessment paths from the knowledge structure [24].

For each running test, Studio collects basic quantitative data about the number of assigned questions, how often tests are taken and how many students open which test and when. This is completed by qualitative measures, collecting which questions and knowledge elements the students passed or failed. To conclude further on the mechanisms and impacts of Studio within the current study, a new logging system was developed, collecting the interaction with the system and detailed information about the feedback as detailed events.

Each event stores information about the system in 7 dimensions, as described in Table 1 below:

Table 1: Event blueprint to store events concerning system interaction.

Attribute	Description
Event description code	Which type of event and what factors are relevant.
Location code	On which part of the assessment-process or interface the event has occurred.
Session identifier	Each access of the system is one session for one user.
Numerical value storage	Multi-purpose field, filled depending on the event type.
String value storage	Multi-purpose field, filled depending on the event type.
Event-time	The time of the start of the event.
Item reference	A unique reference code, identifying the correlated item within the ontology. E.g. a question or a knowledge-element ID.

All events are stored in order of their occurrence, so if no explicit end event is defined, the next event for the same session and user is acting as the implicit end date. Extending the existing storage of information within Studio, the new logging system stores the additional events, as shown in Table 2 below:

Table 2: Assessment events and descriptions.

Event type	Description
START_TEST	Marks the start of a test.
END_TEST	Marks the end of a test.
OPEN_WELCOME_LM	The user opened the welcome page.
OPEN_LM_BLOCK	The student opened a learning material block on the test interface.
OPEN_LM	The student opened the learning material tab on the test interface.
RATE_LM	The student rated the learning material.
CHECK_RESULT	The student opened a result page.
CONTINUE_TEST	The student submitted an answer.
FINISH_TEST	The test has been finished.
SUSPEND_TEST	The user suspended the test.
RESUME_TEST	The user has restarted a previously suspended test.
SELECT_TEST_ALGO-RITHM	The algorithm used to actually test the student is selected.

TEST_ALGORITHM_-EVENT	The behavior of the current test algorithm changes, e.g. entering another stage of testing.
ASK_TESTQUESTION	Sends out a test question to the user to answer.
STUDIO_LOGOUT	The user logs out of the Studio system.

To store the events, the system implements an additional logging database, splitting the concepts of the logging to a star-schema for efficient extraction, transformation and loading. The logging system is modular and easy to extend with new concepts and easy to attach to potential event positions within the Studio runtime. Together with the existing logging of the assessment evaluation feedback, this new extension tracks the exploration of the sub-ontology within the assessment and enriches the feedback data with context information of the students behavior on the system.

4. NEXT STEPS

The domain ontology offers a functional and semantically rich core for supporting learning and education. Yet not all the semantic potentials are fully leveraged to support and test the learner's progress. The "requires_knowledge_of" relation-requirement is a potential start-concept to model sub-areas as groups which together compose the dependency. This could act as an additional input for the assessment, where the system derives more complex decision how to further explore the related parts of the structure [25]. This could also be visualized, enabling the learner to grasp the personal knowledge as a visible group of concepts.

Besides giving colors to the different types of relations, the visualizing of edges between knowledge areas is yet unfiltered, offering no further support for navigation. A next stage of implementation could be the introduction of a visual ordering and grouping of knowledge areas and relations. Underlying relations of sub-nodes could be interpreted visually through the thickness of relations between nodes, easing the perception of complex parts of the domain ontology, especially within administration and maintenance tasks.

The feedback of the current evaluation study of Studio will provide additional insights into the usage of the system by the students. Based on this new data it is possible to mine profiles over time on the knowledge structure. One major application is here the creation of behavior profiles, as proposed in [23].

5. REFERENCES

[1] D. Dunning, C. Heath, and J. M. Suls, "Flawed self-assessment implications for health, education, and the workplace," *Psychological science in the public interest*, vol. 5, no. 3, pp. 69–106, 2004.

[2] D. Jonassen, "Reconciling a Human Cognitive Architecture," in *Constructivist Instruction: Success Or Failure?*, S. Tobias and T. M. Duffy, Eds. Routledge, 2009, pp. 13–33.

[3] R. Vas, "Educational Ontology and Knowledge Testing," *Electronic Journal of Knowledge Management*, vol. 5, no. 1, pp. 123 – 130, 2007.

[4] M. Y. Dahab, H. A. Hassan, and A. Rafea, "TextOntoEx: Automatic ontology construction from natural English text," *Expert Systems with Applications*, vol. 34, no. 2, pp. 1474–1480, Feb. 2008.

[5] S.-H. Wu and W.-L. Hsu, "SOAT: a semi-automatic domain ontology acquisition tool from Chinese corpus," in *Proceedings of the 19th international conference on Computational linguistics-Volume 2*, 2002, pp. 1–5.

[6] M. Missikoff, R. Navigli, and P. Velardi, "The Usable Ontology: An Environment for Building and Assessing a Domain Ontology," in *The Semantic Web — ISWC 2002*, I. Horrocks and J. Hendler, Eds. Springer Berlin Heidelberg, 2002, pp. 39–53.

[7] T. A. Gavrilova and I. A. Leshcheva, "Ontology design and individual cognitive peculiarities: A pilot study," *Expert Systems with Applications*, vol. 42, no. 8, pp. 3883–3892, May 2015.

[8] S. Sosnovsky and T. Gavrilova, "Development of Educational Ontology for C-programming," 2006.

[9] V. Psyché, J. Bourdeau, R. Nkambou, and R. Mizoguchi, "Making learning design standards work with an ontology of educational theories," in *12th Artificial Intelligence in Education (AIED2005)*, 2005, pp. 539–546.

[10] T. Nodenot, C. Marquesuzaá, P. Laforcade, and C. Sallaberry, "Model Based Engineering of Learning Situations for Adaptive Web Based Educational Systems," in *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers &Amp; Posters*, New York, NY, USA, 2004, pp. 94–103.

[11] A. Bouzeghoub, C. Carpentier, B. Defude, and F. Duitama, "A model of reusable educational components for the generation of adaptive courses," in *Proc. First International Workshop on Semantic Web for Web-Based Learning in conjunction with CAISE*, 2003, vol. 3.

[12] W. Chen and R. Mizoguchi, "Learner model ontology and learner model agent," *Cognitive Support for Learning-Imagining the Unknown*, pp. 189–200, 2004.

[13] G. Neusch and A. Gábor, "PROKEX – INTEGRATED PLATFORM FOR PROCESS-BASED KNOWLEDGE EXTRACTION," *ICERI2014 Proceedings*, pp. 3972–3977, 2014.

[14] H. L. Roediger III, "Relativity of Remembering: Why the Laws of Memory Vanished," *Annual Review of Psychology*, vol. 59, no. 1, pp. 225–254, 2008.

[15] J. D. Fletcher, "Evidence for learning from technology-assisted instruction," *Technology applications in education: A learning view*, pp. 79–99, 2003.

[16] J.-C. Falmagne, M. Koppen, M. Villano, J.-P. Doignon, and L. Johannesen, "Introduction to knowledge spaces: How to build, test, and search them," *Psychological Review*, vol. 97(2), pp. 201–224, 1990.

[17] F. Neuhaus, E. Florescu, A. Galton, M. Gruninger, N. Guarino, L. Obrst, A. Sanchez, A. Vizedom, P. Yim, and B. Smith, "Creating the ontologists of the future," *Applied Ontology*, vol. 6, no. 1, pp. 91–98, 2011.

[18] "Ext JS API." [Online]. Available: http://www.objis.com/formationextjs/lib/extjs-4.0.0/docs/index.html. [Accessed: 04-May-2015].

[19] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst, "Animated exploration of dynamic graphs with radial layout," in *IEEE Symposium on Information Visualization, 2001. INFOVIS 2001*, 2001, pp. 43–50.

[20] M. Arru, “Application of Process Ontology to improve the funding allocation process at the European Institute of Innovation and Technology.,” in *3rd International Conference on Electronic Government and the Information Systems Perspective (EGOVIS).*, Munich, Germany, 2014.

[21] M., M., Ansari, F., Dornhöfer Khobreh and M. Fathi, “An ontology-based Recommender System to Support Nursing Education and Training,” in *LWA 2013*, 2013.

[22] V. Castello, L. Mahajan, E. Flores, M. Gabor, G. Neusch, I. B. Szabó, J. G. Caballero, L. Vettraino, J. M. Luna, C. Blackburn, and F. J. Ramos, “THE SKILL MATCH CHALLENGE. EVIDENCES FROM THE SMART PROJECT,” *ICERI2014 Proceedings*, pp. 1182–1189, 2014.

[23] H. M. Truong, “Integrating learning styles and adaptive e-learning system: Current developments, problems and opportunities,” *Computers in Human Behavior*, 2015.

[24] C. Weber, “Enabling a Context Aware Knowledge-Intense Computerized Adaptive Test through Complex Event Processing,” *Journal of the Sientific and Educational on Forum on Business Information Systems*, vol. 9, no. 9, pp. 66–74, 2014.

[25] C. Weber and R. Vas, “Extending Computerized Adaptive Testing to Multiple Objectives: Envisioned on a Case from the Health Care,” in *Electronic Government and the Information Systems Perspective*, vol. 8650, A. Kő and E. Francesconi, Eds. Springer International Publishing, 2014, pp. 148–162.