View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by LJMU Research Online

Research and Practice in Technology Enhanced Learning
Vol. X, No. X (201X) page# start—page# end
© Asia-Pacific Society for Computers in Education

A FRAMEWORK TO SUPPORT AN INTELLIGENT TUTORIAL SYSTEM FOR COMPUTER PROGRAMMING

DHIYA AL-JUMEILY

Applied Computing Research Group,
Liverpool John Moores University, Liverpool, UK
*D.Aljumeily@ljmu.ac.uk **

ABIR HUSSAIN

Applied Computing Research Group,
Liverpool John Moores University, Liverpool, UK
A.Hussain@ljmu.ac.uk

MOHAMMED ALGHAMDI

Applied Computing Research Group,
Liverpool John Moores University, Liverpool, UK
M.Y.Alghamdi@ljmu.ac.uk

CHELSEA DOBBINS

Applied Computing Research Group,
Liverpool John Moores University, Liverpool, UK
C.M.Dobbins@ljmu.ac.uk

JAN LUNN

Applied Computing Research Group,
Liverpool John Moores University, Liverpool, UK
J.Lunn@ljmu.ac.uk

Technology has been used in the last three decades to support teaching and learning. However, educational software has frequently been under investigation to check the validity of their benefits. There is a demand for increasingly intelligent pedagogically-grounded computer technology. In this paper, we discuss adaptive, crowd sourced, and primarily educational technology; targeted at software development students. The proposed technology caters for either individual or group learning. It differentiates itself from other tutoring and programming support technologies as it will continually monitor and assess students' performance in each phase of the educating process. It will also guide them in their learning through interactive feedback and adaptive curriculum delivery that suits both their current levels of learning and preferred learning styles.

Keywords: Technology and Education; Coding; Teaching and Learning; Computer Programming; Adaptive Software.

1. Introduction

Teaching novice programmers the skills associated with software development is a challenging process (Kim & Lerch 1997). This is due to the fact that teachers are required to individually asses their students and then, according to their existing level of knowledge and preferred learning styles, start teaching them a number of tasks such as the technical aspects of programming, new ways of thinking to solve problems and so on. Moreover, programming is essentially a technically-rooted and practical set of skills. Therefore, beginner programmers need to build their skills in entering code, building software, and then as necessary executing, debugging and correcting the software. In practical, lab-based sessions, this often needs one-on-one help from teaching staff. With large class sizes and demands on tutoring staff, weak students in particular may not have the opportunity to get the individual help they require (Wang, et al. 2011).

At the present time, there are no intelligent adaptive or individualized tutoring technologies that satisfactorily solve those above issues. Given the demands on timescales for teachers' (and teaching assistants'), there is a clear benefit to automated software that can assist in actively tutoring students of software development. Therefore, in this research work, we intend to provide a solution of supporting some of those identified limitations. One of our aims of this exploration is to integrate the concept of "assessment for learning" into a learning technology to better adapt student learning capabilities. Furthermore, recognizing and reacting to learners' preferred delivery styles to improve student performance and increase their engagements into learning materials are another aim of this study. Also, this report has in depth analysed some of the issues of the crowd- sourced educational applications. For example, a popular web community, Stack Overflow, is cited as an example of crowd-sourced education. It provides a fast "first answer" response time of on average, 11 minutes; with contributing users rewarded for their participation with a reputation points scheme. Stack Overflow is used by the software development community to share and solve common problems and solutions/suggestions. Its reward scheme encourages contributions while allowing information recipients to judge the perceived quality of the help they're receiving. Educational crowdsourcing applications of this nature help lecturers, students and professionals in communicating with each other; primarily asking questions and receiving solutions. However, there are still some significant limitations in those applications; how they assess the quality of the learning (i.e. was it just copying or at best learning by rote), or considering individual differences amongst learners (Mamykina, et al. 2011).

The layout of this paper is as follows. Section 2 briefly describes the background and general overview of the different learning styles of student's, while Section 3 explores crowd sourcing in education and its applications. Section 4 details the design of the proposed system, whilst section 5 provides a summary of the paper and the future direction of the research.

2. Learning Style

Rutherfoord & Rutherfoord (2008) defined learning styles as the characteristic techniques in which learners learn, understand and get information. Some researchers defined a learning style as an approach of learning a concept. This is because each learner has a different preferred approach to understanding or learning things. For example, some learners prefer to, and perform better when learning visually, while others may prefer to learn aurally (Hawk (2007); Rutherfoord&Rutherfoord (2008)). Considering learning styles of all students in the traditional classroom can be a challenging issue for teachers. Teachers have only a limited time in preparing their materials and delivering their classes, lectures and tutorials (Watson, et al. 2010). Established pedagogical theory specifies several learning style models (Graf, et al. 2007), including Kolb Experiential Learning Theory (Hawk (2007)), the VARK Model (Hawk (2007); Leite, et al. (2009)), Felder–Silverman Learning/Teaching Style Model and Dunn and Dunn Learning Style Model. Moreover, each of these models has different descriptions for the learning style. The planned research centres on the VARK Model as it is the most widely influential model (Eltigani, et al. 2011).Equally, our proposed work will also be evaluated for students who learn software development primarily by doing (Kinaesthetic Learners), and VARK provides most support for this style. The following subsection discusses VARK model in more detail.

2.1. *The VARK Model*

The acronym VARK stands for Visual (V), Aural (A), Read/Write (R), and Kinesthetic (K). Learning style has been defined in this model as a learner's preferred ways of remembering, understanding, and reasoning about knowledge. The VARK model has been used for advising teachers for knowing the preferred learning styles of their students (Eltigani, et al. 2011) Significantly, this model has a supporting validated questionnaire (and website version at http://www.vark-learn.com/english) that allows a reasonably quick (self) assessment of learning style preference. This can be done by filling in the online questionnaire, which then shows the website or allows calculation of the VARK learning style. VARK defines four learning styles (Hawk (2007); Eltigani, et al. 2011), as follows:

a) **Visual**: one of the original basic learning styles. In this particular type, a learner learns best by seeing. For example, flowcharts, diagrams, maps and so on.
b) **Aural**: another significant learning style in traditional classroom education. Here, a learner prefers to learn best through listening to lectures, discussion, tapes and etc.
c) **Read/Write**: These learners prefer self-directed learning – e.g. reading textbooks, reports, or webpages and then summarize or write what they have understood.
d) **Kinaesthetic**: This is another primary learning style in the classroom. Kinaesthetic learners do so best through experience; undertaking experiments, carrying out case studies, practical sessions, etc.

The next section discusses some existing applications of learning style-sensitive software.

2.2. *Learning Styles in Software*

There are several adaptive educational hypermedia systems that, as part of their adaptive process, consider the learning styles of the learners. However, they still have some limitations (Eltigani, et al. 2011). Some of these applications and their limitations are as follows.

2.2.1. *iWeaver*

This is an adaptive tutoring system used to teach the Java programming language. Wolf reported that the aim of developing his system was to accommodate individual learning styles in an adaptive e-learning environment. The learning process inside this system is described in the following steps. First, when a learner logins into the system, the system will request from this learner to answer 118 questions of the Building Excellence Survey. Once the survey is completed, the learner is given an explanation of his/her suitable learning style within some recommendations on the media representation for the first content module. After that, the learner is able to study the first module in his/her preferred learning style or other styles. Once the learner finishes studying, they are given automated feedback by the system (Wolf, 2002). However, this system is missing some of the important aspects of teaching and learning; for example there is no a pre-assessment for the programming level of the learner. Watson et al note that the iWeaver system fails to express any pedagogical meaning beyond a very simplistic representation of the relationships between curriculum elements.

2.2.2. *Protus*

Protus is an adaptive, intelligent web-based programming tutoring system that is also used for teaching Java programming language. Leaner profiles are created with some basic information; then the learner's preferred learning style is ascertained via a set of questions. This information is stored in the profile and used to select the appropriate lesson customisation for the specific learning style (Klašnja-Milićević, 2011). However, this system does not provide any significant functionality towards adapting curriculum towards learner ability; there is no assessment-driven learning, nor any initial diagnostic assessment. In order to create a truly adaptive system, the learner's current – and developing ability – must be tested.

2.2.3. *AEHS-LS*

AEHS-LS, or Adaptive E-learning Hypermedia System based on Learning Styles, was used for teaching the Javascript scripting language. The authors state that it was designed to

assess the consequences of adapting educational materials individualized to the student's learning style. As with the Protus system, learners create an associated profile during the registration process. Again, the learner is responsible for selecting their appropriate learning style. AEHS-LS prompts the user to select their own learning style, if known, and if not, prompts them with the Fleming VARK questionnaire. Once the learning style is either determined or selected, lessons are then delivered according to the selected style. AEHS-LS defines a strict outcome-concept structure, such that lessons follow a traditional structure outlining concepts, delivering materials, and then summarising with a plenary. Appropriate style-specific resources are generated for each concept by a subject expert and then simply selected by the software at delivery time. Responses to plenary quizzes are used to monitor performance against a particular style, continually adapting the selected learning style. It is not clear from the AEHS-LS published work exactly how the learning style adaptation assesses performance against alternative styles.

Analysis of the resulting system showed that AEHS-LS-engaged students outperformed the control group students. However, student feedback suggested that the auditory learners experienced difficulty; though this is not attributed to the system's approach. It is suggested that this is due to audio delivery in a language other than the participants' native language (Eltigani, et al. 2011). The AEHS-LS study does not investigate this further.

The research work in developing these systems has clearly conducted valuable investigations into harnessing technology as a mechanism for adapting curriculum / delivery in accordance with a learner's preferred style. Equally, they appear to demonstrate, in limited evaluations, that correctly exploiting a learning style does improve assessment performance. However, it is clear that the systems do not fully address either the pedagogical or technical concerns regarding learning-style-adaptive learning support systems.

Here is to summarize some of the significant missing pedagogical impacts around the above discussed applications. They are: "iWeaver", "Protus" and "AEHS-LS". Those applications have not considered what learners need to be taught as there is no a diagnostic assessment for them. Another shortcoming would be that the differences among learners have not been thought over in those technologies. The following section further investigates the interaction between learning styles and technology.

3. Technology and Learning Styles

This section discusses the interactions between technology and learning styles. The first subsection looked at how pedagogical research and practice in learning style mapping and application can be applied to existing technological approaches. The second subsection examines the potential for technological impact to augment existing pedagogical practice. The final subsection discusses criticism of learning styles – both in the classroom and in e-learning environments.

3.1. *The Impact of Learning Style in Technology*

Learning styles have several potential areas of impact in existing technology. One such impact is utilisation of data about learning styles to improve the quality of e-learning systems' adaptation models. Intelligent e-learning systems should ideally track a learner's progress, and optimise the learning process to take advantage of a learner's strengths and help them to overcome their weaknesses. There is evidence from recent studies that suggests students who engage with a system that incorporates a learning style track-and-response mechanism outperform those who study outside the system. For example, 70 % of students who used the Protus system to learn Java found this adaptive system successfully guides them to the appropriate materials with useful explanations (Klašnja-Milićević, 2011).

Another potential impact of including learning styles in learning software is helping to personalise the learning experience – and importantly, increase engagement. Several of the educational technology systems were designed to suit a variety of learning styles for learners ((Klašnja-Milićević, 2011;Wolf, 2002). The vast majority of students engaged in these systems found e-learning systems are more enjoyable than the traditional learning system in the classroom. One significant advantage in this regard is that a well-designed software system can make these identifications and selections with little computation cost; contrasting with teacher effort to correctly identify and respond to all of the learners and their differing learning styles in a large classroom (Watson, et al. 2010).

3.2. *The Impact of Technology on Learning Style*

Just as good pedagogical practice can feed into the design of tomorrow's e-learning systems, technology can continue to feed back into teaching practice. For example, lecturers already engage their students more thoroughly through the use of additional multimedia content (Stickel, 2009). Additionally, technology provides a means to reach a wider range of students (Wolf, 2002). However, there is a significant advantage of e-learning - in terms of the potential for increasing teaching and learning output; letting subject experts focus on material creation, and automating much of the repetitive tasks. Deferring time-consuming tasks to a software system allows greater one-on-one teaching and learning time; a challenging prospect in the traditional classroom (Wolf, 2002).

Technology can help to rapidly assess many learners' learning styles. For example, iWeaver determines the learning style of their users by asking them over 100 multiple choice questions, with the system automatically providing the content in their preferred learning styles.

However, technological tools do not yet suit all of the types of learning styles. This is due the fact that teaching materials are not always adaptable to all types of learning styles. Put simply, some topics do not lend themselves to all the VARK styles. Equally, certain kinaesthetic learning tasks are ill suited to an electronic or virtual environment. For example, Tablet PC is a teaching tool used in engineering courses. Kinaesthetic learners

evaluated this tool as un-engaging, while visual learners found it an enjoyable classroom addition, and they have a greater preference for it (Stickel, 2009).

3.3. *Learning Styles Criticism*

By looking at the previous studies around introducing learning styles in adaptive E-learning hypermedia system, there is a big debate. Yasir Eltigani found that including learning styles in E-learning hypermedia system helped to improve students' achievement and performance (Eltigani, et al. 2011). Conversely, Brown et al reported that there is no evidence to support the idea that matching learning styles to learners improves learning effectiveness; although their sample was primary school children. Elvira Popescu criticised the learning styles approach for several reasons. One complaint is that there is a large number of learning style models, with no unanimously accepted approach. Additionally, the length of the assessment questionnaires was considered to discourage participants. Popescu suggested that learning style questionnaires should be revised for use in web-based learning systems as they ignore technology related preferences.

Additionally, the authors of this paper have identified other issues in integrating electronic-selected learning styles into teaching. One significant issue is that of teaching workload, particularly for those tutors tasked with creating their own materials. Designing several sets of much the same material; each tuned to a particular learning style is likely to be very time consuming, requiring a considerable increase in effort.

Another issue would be that some subjects are naturally not suitable to be taught in accordance with a particular style. For example, teaching heavily verbose mathematics or programming subjects would be very difficult to engage the auditory learners. Also, developing materials for auditory learners may create other challenges as a student's language may differ from the delivery language. Yasir Eltigani noted that his auditory learners who natively speak Arabic found that listening to spoken English by a non-native is difficult (Eltigani, et al. 2011). Therefore, those above discussed issues should be taken into a consideration by a close future study. More importantly, on this report we aimed not only to discuss those issues, but we also aim to consider them into our planned automated system and trying to tackle as many of them as possible.

4. Crowd Sourcing In Education

There are a wide range of definitions that can be applied to the term "crowdsourcing". One of those definitions is that crowdsourcing can be an online community facilitating a large group of people from across the globe to meet with each other in order to discuss and exchange ideas, solve problems, and share entertainment (Brabham, 2008). However, Estellés-Arolas et al refine and integrate this definition of crowdsourcing to say that it is a "participative online activity" including a group of community stakeholders of "varying knowledge, heterogeneity, and number" aimed to achieve a task through volunteer labour (Estellés-arolas & González-ladrón-de-guevara2012). We will return to this notion of volunteer-supplied labour and how participants are motivated later in the section.

There are several subtypes of crowdsourcing. The types we consider here in the context of education are: crowd creation, crowd voting, crowd funding and crowd wisdom (Brabham, 2008). Crowd funding has an interesting potential application in education; its popular application thus far includes targets such profit-driven commercial debt distribution through to more altruistic projects such as disaster relief, or ethics-driven arts support – e.g. label-less movie and music production. While crowd-funding may have a future role in research-led teaching and learning, our focus here will be on crowd voting and crowd wisdom. This is because in the proposed architecture, we intend to utilise these two features. Whilst not a new pedagogical concept; i.e. "hands up if", or the anonymised modern equivalent of Clickers, crowd voting techniques can easily be applied to technology-enhanced learning. It can help to collect and gauge a large crowd's view on a certain area of learning; for example, if we are gauging satisfaction of learning materials or informally checking understanding of a learning outcome. Equally, crowd wisdom allows the aggregation of data in the form of problem solutions and sharing exercise workings between "crowds" of students. However, providing a technological solution is not necessarily in itself sufficient to produce these effects. Thus, beyond the desired collegiate effect of a cohort of students engaged together in study, engaging the "crowd" in participation is a key to the success of these technologies. Three popular techniques are often identified to attracting a crowd (Brabham, 2008;Franklin, et cl. 2011). The first is simply to financially reward crowd participants; awarding money or exchange points for contributing to votes or giving information. Secondly, communities desiring crowd-action may provide entertainment to attract and retain crowds. Participants may receive a game, music, a film – as reward for contribution, or the entertainment may simply be used to attract them to the crowd.

Finally, and perhaps most related to educational crowdsourcing, is the altruistic participation or community reward. Participants join the crowd for the reward of participating; the exchange of information. This is perhaps a more obvious draw when it comes to rewarding novice students – they will benefit from good quality information in the form of solution assistance and guided discussion. However, beyond the collegiate-like effect of gifted and talented students feeling personal satisfaction in helping less able students, it is not immediately obvious as to how these high-ability students can be attracted into participation. As such, investigation and experimentation must be conducted into the value of other reward systems for able student crowd participation. The next section explores some current crowdsourcing communities and applications; focusing on those with a primarily educational purpose.

4.1. *Educational Crowdsourcing Applications*

There are several existing crowdsourcing applications and communities used as online education support tools (Buecheler, et al. 2010). An overview of several examples, illustrative of the types of crowd-sourced education, is shown in table 1. They are then expanded on in the rest of the section.

Table 1: Crowdsourcing Application Comparison

App	Overview	Crowd Type	Challenges
Stack Overflow	Used to exchange questions and answers in topics of software development and programming.	Wisdom / Voting	Unsuitable for novice programmers. Democratic-only assessment of quality.
Wikipedia	It is a wiki-based web application, which allows people to add, modify, or delete content in collaboration with others.	Wisdom	Scientific assessment to assess credibility of crowd inputs. They are only assessed "democratically"
CourseEra	An educational, technological organization that offers free online courses in a wide range of topics to teach millions of students rather than hundreds.	Crowd creation	No long term engagements for students. No collegiate support.

"Stack Overflow" is a crowdsourcing community application used in the education. It is used to exchange questions and answers between programmers on a wide range of topics in computer programming. It can help professional programmers to get a quick answer for their questions. However, it has several limitations as an educational tool. For example, it is not suitable for novice programmers as they need a unique level of detailed feedback (Chen, etc. 2012;Antonio 2013). More importantly, there is a distinct pattern describing who often contributes to the site and why they have a contribution. Some studies has reported most contributions in Stack Overflow come from constantly active contributors (Antonio 2013); whereas infrequent users post more queries than give answers. As such, this suggests that the motivation for those active – and answer-providing - contributors would be both altruism and the "reward" of non-currency reputation points. However, the altruism and community engagement may be helped as when active contributors earn more reputation points, they gain site privileges; such as voting to delete answers.

"Wikipedia" is another example of crowdsourcing information. However, it can also be argued that Wikipedia is an online crowd-sourced education repository. There is no formal application process to gain edit privileges on Wikipedia; in fact most pages can be edited anonymously; save for the user's IP Address. Users who choose to register are not required to state or validate their experience or qualification background. It does not have an assessment tool that assess the inputs of the editors (crowd wisdom) to indicate whether their inputs are credible or not (Weld et al. 2012). Conversely, due to its massive user base, Wikipedia does have many advantages; it is a growing multilingual platform, and content is provided democratically, by consensus. While this in itself does not guarantee correct

information – and could be potentially hostile to new research – there is a growing "citation needed" culture. This is where page editors may mark a document at a contentious or doubtful paragraph to request a citation to help readers verify whether the content is verifiable or not. Returning to the edit-by-consensus model, while this can stifle new work, it is still an important pedagogical aspect - peer-review. The good thing about Wikipedia's peer-review model is that anyone with access to the Internet can edit the work of others or request a citation needed on information they do not thing is common knowledge. Research has shown that Peer-review would improve the quality of published works by for example identifying scientific mistakes or wrong references (Weld et al. 2012). However, extreme care should still be employed when using Wikipedia as a reference material. There is a hierarchy whereby pages can be protected and edited only by authorised users. While the authorised user hierarchy is a loosely democratic structure, it is also corporatized – it receives criticism for being nepotistic, unequal and inconsistent in its application of rules.

"Courseera" is a recent examples of a MOOC (Massive Open Online Courses) (Walker, L. 2013). It is the product of an educational organization that offers free online courses in a wide range of topics to teach millions of students rather than hundreds. The Courseera organisation partners with top universities (Crowd creation) in the world to offer online courses for free to anyone. It is still an immature market offering and its revenue model is not yet clear; thus its underlying ethos, whilst appearing altruistic, may still yet evolve or become marketised. Even at this early stage, several challenges have emerged. The first is the course retention rate; some students withdraw from the online courses at an early stage of the course. Weld et al noted that under 15% of students completed the Norvig/Thrun online AI class and out of 104,000 students registering for Stanford's 2011 Machine Learning class, 46,000 submitted the first assignment, and 13,000 passed (Weld et al. 2012).

This section has discussed in depth a number of crowd-sourced applications. They are: "Stack Overflow"," Wikipedia, and "CourseEra".Those applications offer some advantages to the public, for instance, getting a quick answer for such programming posted question and giving a learner more a freedom space to participate and give their own opinions. However, they still have several drawbacks. For example, "Stack overflow" was considered as an unsuitable application for novice programmers. This is because learning quality is poorly managed, and also there is no streaming of ability - differentiation between the expertise level of a professional and a novice programmer. Moreover, another challenge within the above application was student engagements. Therefore, any proposed future technology-enhanced learning system must look to understand and address the issues behind poor retention (and the implied measure of engagement). For example, better engagement in an automated course system may be achieved by better differentiation of learner ability. Previous work by the authors of this paper has explored learner differentiation by assessment for learning (Alghamdi, et al., 2013).

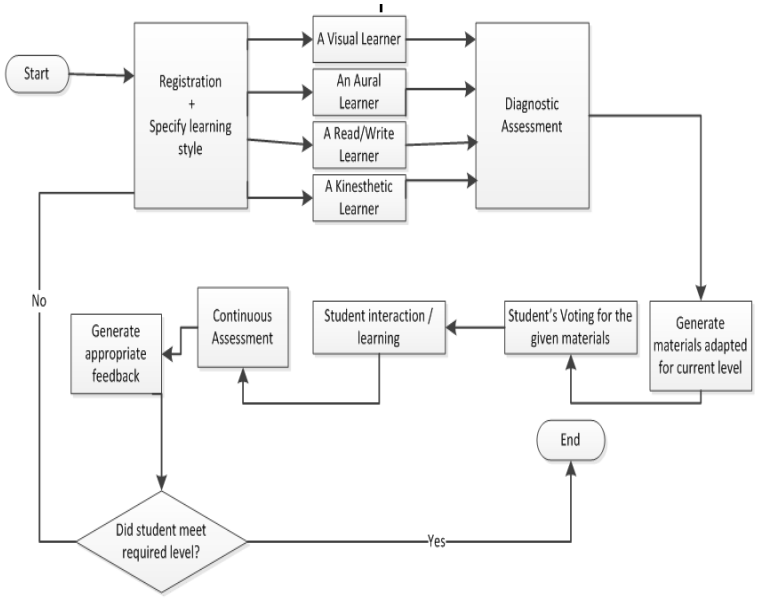
5. Teaching, Learning and Assessment for an Intelligent, Adaptive Learning System

The main inspiration for the proposed system is the Assessment for Learning (AfL) initiative, comprising diagnostic and continual assessment as well as linking AfL with the preferred learning style of the learner.

This defines a structured learning approach based on a student's prior knowledge and student's learning style preference, followed by learning informed by a student's assessment performance. This methodology is applied in the proposed system, such that curriculum sequencing and material generation is fully integrated into an adaptive, student-centric learning tool. This process is shown in Fig. 1.

Figure 1: Learner-Followed Process In Proposed System

When first time learners enter this proposed system, they need to sign up to the system by using a registration form. Once a learner registers, a learner profile will be created to



store all their information and will be saved in the Student Knowledge model. On this proposed system, the VARK learning style model is employed, as it is one of the most influential and flexible models (Eltigani, et al. 2011). When the registration is done, the system will show the learner a short tutorial that explains the four styles in the VARK learning style model.

After showing the tutorial, the system will do two significant diagnostic tasks with the learner. The first task is to determine the learner's preferred style by asking them to fill in the online VARK questionnaire. This information is logged in the student's knowledge model. The following task is to test the learner's prior knowledge of the subject via Diagnostic Assessment; establishing the entry level of ability (see Fig. 2).

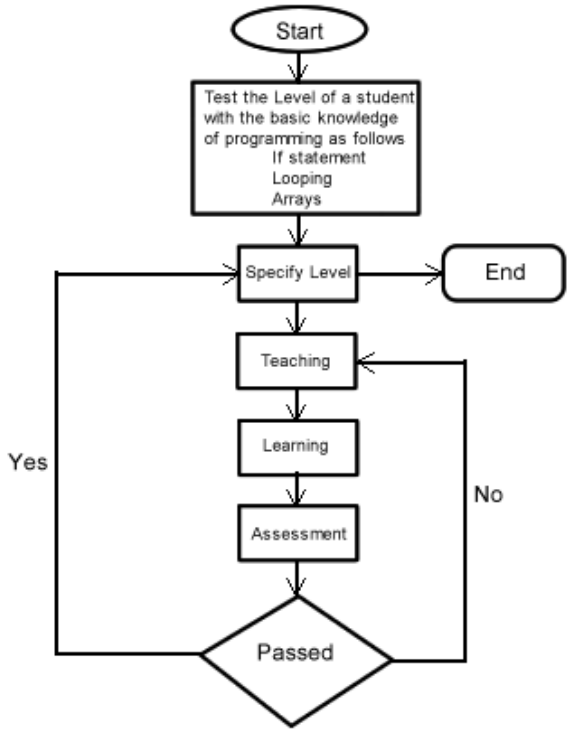


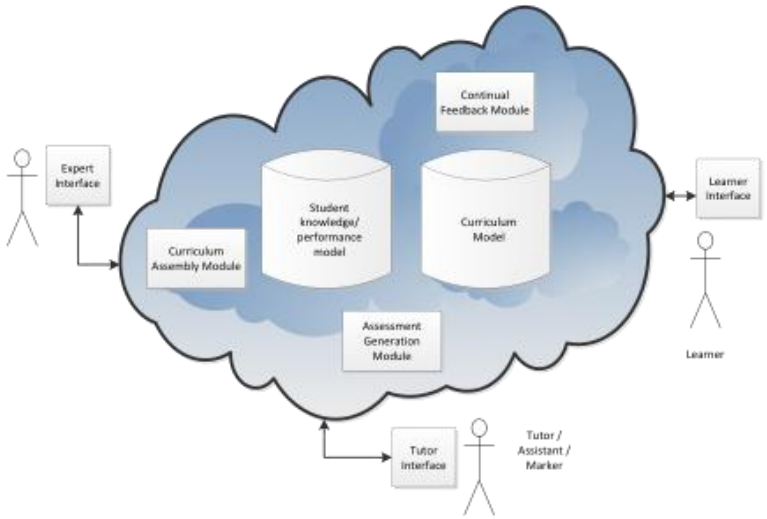
Figure 2: Data flowchart of the proposed system

Upon completing this test, the system will specify the current level (Beginner-Intermediate-Advanced) of this student and then direct him/her to the next step (Teaching). In this step the system will generate the appropriate curriculum material (suitable at entry level and transposed to the correct learning style) for the learner. Then, the system will take him/her to the "Learning" part of this phase, where he/she will start doing some more programming examples with intelligent help on each step of problem solving, including giving a hint to executing the next step. Following to that, the system will take him/her to the "Assessment" part as in this step he/she will have the most appropriate exam which can range from a simple question to a complex programming problem as well as the system will show his/her code error with providing error feedback for that. Lastly, in case this student passed the assessment part, the system will specify a new level for him/her (until he/she achieves the advanced programming level).

The system will allow the learner to vote for the material that has been provided in order to see whether they are satisfied or not. The student continues to be engaged in formative "Continuous Assessment", providing appropriate feedback and adapting the curriculum and learning styles appropriately. While the system framework architecture is a matter for the research work itself, it is envisaged at this early stage that the system will comprise a number of modules that use a variety of artificial intelligence techniques to interpret models defining things like the curriculum being followed (i.e. teaching materials and intended learning outcomes), the pedagogical aspects of this curriculum (e.g. appropriate assessment methods), and individual learner performance aspects (e.g. assessment results).

5.1. *Architecture of the Proposed Adaptive Tutoring System*

The architecture of the proposed adaptive tutoring system is concerned with modelling and interpreting the data discussed over the previous sections to fulfil the pedagogical needs identified. A high-level overview of the architecture is shown in Figure 3.



T

Figure 3: Intended high-level architecture showing example data models

The system comprises of two distinct knowledge models; the student knowledge / performance model, and the curriculum model. These models are designed to maintain information on: a) current student knowledge, sourced from assessment performance, and b) curriculum knowledge, such as learning materials and outcomes, along with assessment methods. Distinct modules communicate with one or both of these knowledge models, responsible for assembling and structuring learning, generating assessment materials,

feedback and regenerating curriculum materials. These modules are described in more detail in Table 2.

Table 2. Component/Module Overview

Component	Description
Student knowledge model	Contains information about individual student's learning – ranging from materials studied, assessments taken, through to assessment results, extrapolated to identify performance against learning outcomes defined in the Curriculum model.
Curriculum model	Stores curriculum-related data; at its lowest level, specifications of the learning outcomes that make up a unit with differentiation levels. This model is likely to maintain appropriate learning materials and assessment templates for relevant outcomes. This will be populated by tutors and experts to specify courses and modules.
Curriculum assembly	Adapts curriculum model data to produce a series of materials for a given set of learning outcomes, tailored to a specific student model.
Assessment generation	Transforms curriculum model data into appropriate assessments for either a set of learning outcomes provided either directly by a tutor or inferred from a student model's outstanding learning outcomes.
Continual feedback	Produces feedback on assessment submissions, aligning student performance against learning outcome s in the curriculum models, using data from prior student attempts and ongoing tutor input.
Tutor, Learner and Expert interface	Provides a user interface and access control for the various roles. The tutor and expert interfaces will provide intuitive mechanisms for inputting curriculum materials and manually checking assignments and student performance, while the learner interfaces will provide rich lecture, tutorial and assignment user interfaces.

In order to overcome some of the discussed limitations of existing shortcomings of educational crowd sourced systems, the system will incorporate three different platforms. The first one will be for novice programmers. The second platform will be allocated to those who have an intermediate level, whereas the last platform will be for learners who have advanced programming skills. The diagnostic assessment tool will be used to appoint a learner to a suitable platform. In this instance, every learner will be asked to take the diagnostic exam and according to his or her performance, he or she will be assigned to the appropriate platform. For instance, platform number one will focus on the lower learning outcomes whereas the other two platforms will be designed for the intermediate and higher learning outcomes. In all these planned platforms, learners will be given the opportunity to vote for the given material and also learners who actively participate in their specified platforms will be visibly distinguished (for instance given higher privileges to add-delete) from those who do not participate very often.

Furthermore, the proposed system will consider the arrangement of student's activities in those platforms. For example, students would be allowed to advise each other when they

are solving a programming problem that would be weekly generated by our proposed system; however, they will not be permitted to directly provide the answer as the aim of this planned room is to let all students think as one team and give everyone enough time in thinking how to solve a problem. At the end of the week, the system will post the model answer for that problem. Furthermore, other challenges that will be considered include 1) investing how segregating on ability might adversely affect crowd sourcing, 2) the adverse effect that a monitoring system may have on student engagement and 3) how it can be applied to the process of reflective learning into these planned crowd-sourced platforms. The outcome of this stage is supporting our system by the two crowd-sourced tools.

6. Summary and Future Work

Currently, education provision is encountering new challenges, new inventions and emerging technologies. The main area where innovation is presented lies on instructional methodologies that can be presented to open communities. As such, modern education benefits from developments in intelligent systems and widespread high-speed Internet access. Adaptive, intelligent hypermedia is increasingly gaining ground as a pedagogical delivery method, yet still has far to go in terms of refining quality of materials, student performance and engagement monitoring. Furthermore, there are challenges that stem from teaching such complex courses in a distributed manner, which requires an intelligent system.

This paper has discussed significant issues around educational crowd-sourcing applications and intelligent tutoring systems. It has also outlined the plan to tackle pedagogical concerns and how the individualized adaptive, and crowd-sourced technology for software development learners can be developed. Consequently, the proposed framework is an intelligent programming tutoring system that can be used as a learning tool for learning program and is especially useful for novice programmers.

Future work aims to develop the system into a fully functional system that will be evaluated used first year undergraduate students on the "Introduction to Programming" module. Those students will be divided into two groups. The first group will be an Experimental Group (those who will be taught by using the proposed system), and the second group will be the Control Group (those who will be taught in the traditional way such as in a classroom). Three different comprehensive exams will then be used to evaluate the performance of those two groups and will be marked by the proposed system. The control group students will be given sufficient training time in the research system before undertaking the evaluation. However, this student group will have the option, if preferred, to undertake the evaluative exams in paper form; marked by a human teacher. Once those three exams are done, a comparison between both groups' achievements will be made to see the learner level in each of those two groups. This evaluation will provide comparable results and clearly specify which group is performed better. Thus, the outcome of this stage would be the ability to confirm whether the proposed technology is an effective tool for teaching and learning software development students or not.

References

Kim, J., & Lerch, F. J. J. (1997) "Why Is Programming (Sometimes) So Difficult ?," Information Systems Research, vol. 8, no. 1, pp. 25–50.

Wang, T., X. Su, X., Ma, P., Wang, Y. and Wang, K. (2011, Jan). "Ability-training-oriented automated assessment in introductory programming course," Computers & Education, vol. 56, no. 1, pp. 220–226.

Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G. and Hartmann, B. (2011) "Design lessons from the fastest q&a site in the west," Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11, p. 2857.

Chen, Y., Hsu, C. Y., Liu, L. and Yang, S. (2012, Feb) "Constructing a nutrition diagnosis expert system," Expert Systems with Applications, vol. 39, no. 2, pp. 2132–2156.

Rutherfoord, R. H. and Rutherfoord, J. K., 2008 "Exploring teaching methods for on-line course delivery-using universal instructional design," Proceedings of the 9th ACM SIGITE conference on Information technology education - SIGITE '08, p. 45.

Hawk, T. F., 2007 "to Enhance Student Learning," vol. 5, no. 1, pp. 1–19.

Watson, C., Li, F. W. B.and Lau, R. W. H. 2010 "A pedagogical interface for authoring adaptive e-learning courses," Proceedings of the second ACM international workshop on Multimedia technologies for distance leaning - MTDL '10, p. 13.

Graf, S., Viola, S. R. and Leo, T. 2007 "In-Depth Analysis of the Felder-Silverman Learning Style Dimensions," vol. 40, no. 1.

Leite, W. L., Svinicki, M., and Shi, Y. (2009, Aug.) "Attempted Validation of the Scores of the VARK: Learning Styles Inventory With Multitrait-Multimethod Confirmatory Factor Analysis Models," Educational and Psychological Measurement, vol. 70, no. 2, pp. 323–339.

Eltigani, Y., Mustafa, A. and Sharif, S. M., 2011"An approach to Adaptive E-Learning Hypermedia System based on Learning Styles (AEHS-LS): Implementation and evaluation," vol. 3, no. January, pp. 15–28.

Wolf, C., 2002 "iWeaver : Towards an Interactive Web-Based Adaptive Learning Environment to Address Individual Learning Styles," pp. 1–14.

Brabham, D. C., (2008, Feb) "Crowdsourcing as a Model for Problem Solving: An Introduction and Cases," Convergence: The International Journal of Research into New Media Technologies, vol. 14, no. 1, pp. 75–90.

Estellés-arolas, E., and González-ladrón-de-guevara, F. 2012 "Towards an integrated crowdsourcing definition", Journal of Information Science, no. X, pp. 1–14.

Franklin, M. J. Berkeley, U. C. Kraska, T. and Xin, R. 2011 "CrowdDB : Answering Queries with Crowdsourcing," pp. 61–72.

Buecheler, T. Sieg, J. H. Füchslin, R. M. and Pfeifer, R. 2010 "Crowdsourcing , Open Innovation and Collective Intelligence in the Scientific Method : A Research Agenda and Operational Framework Why Crowdsourcing in the Scientific Method," pp. 679–686, 2010.

Antonio, S. 2013 "Facilitating Students ' Collaboration and Learning in a Question and Answer System," pp. 101–105.

Antonio, S. 2013 "Preliminary User Study for Gratitude and Reciprocity in a Q & A System," pp. 169–174.

Weld, D. S. Adar, E. Chilton, L. Hoffmann, R. Horvitz, E. Koch, M. Landay, J. and Lin, C. H. 2012 "Personalized Online Education — A Crowdsourcing Challenge," in Association for the Advancement of Artificial Intelligence.

Walker, L. 2013 "MOOC: Massive Open Online Course Guide"[Online]. Available: http://personalweb.about.com/od/onlineeducation/a/Mooc-Massive-Open-Online-Course-Guide.htm.

Alghamdi, M., Lamb, D., Al-Jumeily, D., Hussain, A. (2013, Dec) "Assessing the Impact of Web-Based Technology on Learning Styles in Education", 6th International Conference on Development in eSystems Engineering (DESE13), IEEE Computer Society, Abu Dhabi.

Klašnja-Milićević, A., Vesin, B., Ivanović, M., and Budimac, Z., (2011, Apr) "E-Learning personalization based on hybrid recommendation strategy and learning style identification," Computers & Education, vol. 56, no. 3, pp. 885–899.

Stickel, M. (2009, Oct) "Impact of lecturing with the tablet PC on students of different learning styles," 2009 39th IEEE Frontiers in Education Conference, pp. 1–6.

Wolf, C. 2002 "iWeaver : Towards ' Learning Style ' -based e-Learning in Computer Science Education," vol. 20.