Scaffolded, Scrutable Open Learner Model (SOLM) as a foundation for personalised e-textbooks.

Judy Kay[1 [0000−0001−6728−2768]] and Bob Kummerfeld[1 [0000−0002−6046−6393]]

University of Sydney, Sydney NSW, Australia,
Judy.Kay@sydney.edu.au
Bob.Kummerfeld@sydney.edu.au
http://sydney.edu.au/

Abstract. While e-textbooks have become increasingly widespread, they have made little use of personalisation. This paper presents an architecture for making a learner's own lifelong learner model as the foundation for a highly flexible form of personalisation for learners using e-textbooks. The architecture has the following core components: an arbitrary *e-texbook platform* that minimally provides log data about the learner; a *personal user model*; an *open learner model (OLM) interface* that provides personalised scaffolding to support reflection on learning progress and support for the learner to scrutinise their model. We present a worked example of the approach, explaining design decisions in terms of a substantial body of previous work on open learner models and support for meta-cognitive processes of self-monitoring, self-reflection and planning. Our key contribution is a new approach to personalised e-textbooks based on the learner's own long term learner model.

Keywords: e-textbook · personalisation · personal learner model · OLM.

1 Introduction

Learners in tertiary education increasingly have ubiquitous access to smartphones and many also own portable larger screen devices that cross the tablet-laptop continuum as well as desktops. This makes it feasible to make use of electronic textbooks since they can be available to the students. This creates opportunities for creating textbooks that are central to the learning in a university subject. It also is timely to create e-textbooks that are far more than simply an electronic form of a conventional textbook using tools such as Runestone [13, 14] which make it feasible for a teacher to create or customise their own e-textbook that includes embedded interactive learning activities such as multiple-choice questions, short answer questions and programming testing. These e-textbooks also make it easy for a teacher to reuse some of the many high quality materials openly available on the web, simply by linking to them.

This paper presents our architecture for a new class of such textbooks that makes use of the rich data from learners interacting with the book. The key

innovation is the creation of an independent Open Learner Model (OLM) [10, 6]. This is a representation of the learner's current progress through the Knowledge Components (KC) of the course. This enables a learner to see a high level view of the teacher's curriculum design, with an indication of their progress on each KC. We call this a Scaffolded, Scrutable Open Learner Model (SOLM). The *scrutability* means that we designed it, from its foundations, to ensure that a learner can scrutinise all the key aspects of the model. From a learner's perspective, this means that the scrutiny interface enables the learner to see: the details of the *evidence* used to infer whether they know a concept or not; the description of the *resolver* which interprets the evidence; and a control mechanism that enables the learner to select different resolvers, for example to set their own standard of performance on self-text quizzes to determine whether a particular Knowledge Component is treated as known or not [12].

The idea of an OLM has a long history, as reviewed in [4] and the idea of making this available to students to guide their navigation was used even in early intelligent tutoring systems, such as ELM-ART [3]. The first use of an independent OLM and of scrutability also have a long history [10]. Similarly, there has been long recognition of the importance of OLMs to support learners in taking responsibility for their own learning because it supports their key meta-cognitive processes of self-monitoring, reflection and planning their learning [11, 5]. Our approach also scaffolds the OLM to help the learner in thee challenging meta-cognitive tasks. This is important for two reasons. First, we want to ensure practical scrutability, meaning that the learner actually can scrutinise their OLM successfully to answer core questions about it. Secondly, there is ample evidence that learners benefit from scaffolding to help them reflect and plan [2, 15].

A final key element of our work is the commitment to a deeply human-centred approach to giving learners control over algorithmic systems. There has recently been broad recognition of the importance Fair, Accountable, Transparent machine learning (FAT*ml)[1]. We consider that learning contexts make these issues even more pressing since we want to create e-textbooks that are deeply based on ensuring learners can access and harness their own learning data and can both scrutinise and control its use in an OLM that shows them their learning progress based on data from an e-textbook.

2 SOLM Architecture

Figure 1 shows the architecture of our system. The core idea is that we use an available e-texbook platform and extract data from each learner's activity to create the evidence in their personal learner model. The arrow from the e-textbook to the learner model indicates a program that processes the learner data, transforming it and adding evidence to the model. The teacher curriculum model at the right represents the teacher's intended progress for a *plausibly ideal student* [7], one who does activities as scheduled by the teacher and performs well,

[1] http://www.fatml.org/,https://fatconference.org/)

but not necessarily perfectly, on the e-textbook activities. The architecture could also extract whole class data to add comparative information of the student's progress compared with the class [8]. At the bottom of the figure is the OLM interface. This will enable the learner to navigate their learner model to scrutinise any part of the model and to select the resolver that interprets the evidence about learning. The learner model will also represent the learner's use of the OLM and this will be used to personalise scaffolding [9, 15].

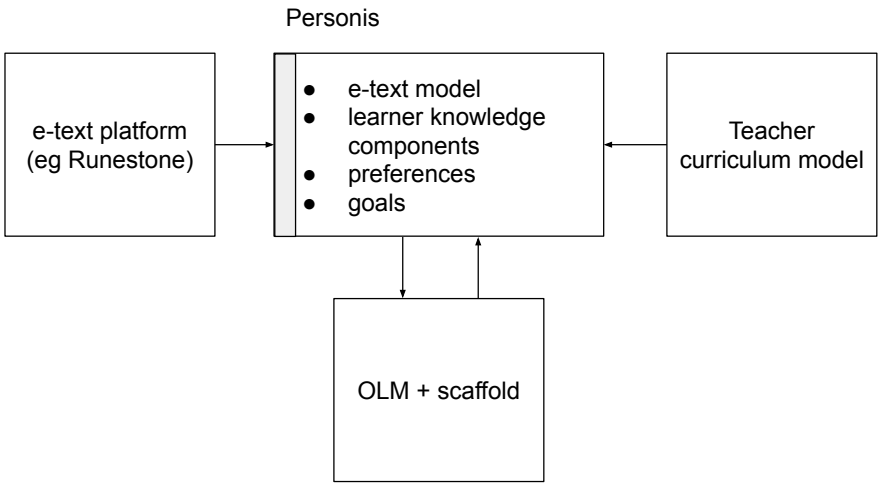

Fig. 1. SOLM Architecture

In order to test our ideas for textbooks of the future we chose the Runestone [2] platform. Runestone is a web based system for constructing interactive textbooks. It allows content to be created in the "restructuredText" markup language and has extensive facilities for inline interactive content such as multiple choice quizzes, fill in the blanks questions, Parson's problems and more. It also allows program code to be written and executed directly in the textbook interface.

The Runestone server is written in Python with the user interface implemented in HTML/Javascript/CSS. The server stores all user data in a database with extensive logging of user actions. Log entries are created when a user loads a page, uses an interactive feature such as multiple choice question or Parson's problem, edits or runs some program code, and others.

For our e-textbook prototype we plan to modify the Runestone system to include API calls on the Personis server to add information to the user model.

[2] http://runestoneinteractive.org/

We prefer to make minimal changes to the e-textbook platform with a view to instrumenting other e-text platforms in the future. The minimum requirement is that the platform be capable of logging activity to a remote server with a simple http request. The sort of activity to be logged includes: fetching a page of the book, answering multiple choice questions, attempting Parson's problems (rearranging lines of text to get the answer).

We do not plan to customise or personalise the e-text itself. Instead the logged activity will be used to drive a scaffolding interface that will guide the user through the e-text.

The architecture diagram illustrates this. The e-text platform sends activity information to our user modelling system(Personis) via a thin layer (shown in grey) that maps e-text elements onto the user model ontology. The user model contains rules that fire when incoming log information cause a significant change in the resolved value of model components and send information to the Open Learner Model and scaffolding interface.

3 Worked example and design rationale

We briefly outline plans for creating an e-textbook for a subject on Human-in-the-loop Data Analytics. This starts with the definition of the learner model hierarchy of contexts and components. The top level should be small enough for learners to readily see their high level progress. For example, our top level has:

- Ethics, privacy, anonymisation, provenance, security and access control
- Human aspects of raw data
- Data collection
- Data cleaning and interpretation
- Processing, machine learning
- Exploration and interpretation interfaces
- Reporting

Within these topics are finer grained components. Ideally, the e-textbook would have interfaces to support easy linking of the content to knowledge components [1] but we will start by simply post-processing of the log data to create evidence in the model. The OLM will also have links back to the relevant parts of the e-textbook, based on the evidence in the model as in MOOClm [7].

4 Discussion and conclusions

Our proposed approach has the following core concepts:

Core role of the Open Learner Model: This offers a light-weight way to provide personalised information about learning progress and a foundation for planning learning. It avoid any need to alter the actual e-textbook and should be able to integrate learning data from additional sources such as an LMS that has data about results of graded assessments.

Need for scaffolding: Since the OLM is intended to play an important role in supporting learner reflection, progress monitoring and planning learning, we see the need for personalised support to help the learner see and understand their model.

Our architecture is based on a vision for each student to have a personal long term scrutable user model. This can provide a unified interface to support their meta-cognitive processes of reflection, self-monitoring and planning their learning in a subject. This is based on the teacher using an e-textbook platform to create a coherent set of resources that include teacher-authored material, links to online materials used as readings as well as self-test and assessment materials.

References

1. Apted, T., Kay, J., Lum, A.: Supporting metadata creation with an ontology built from an extensible dictionary. In: International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems. pp. 4–13. Springer (2004)
2. Azevedo, R., Hadwin, A.F.: Scaffolding self-regulated learning and metacognition–implications for the design of computer-based scaffolds. Instructional Science **33**(5), 367–379 (2005)
3. Brusilovsky, P., Schwarz, E., Weber, G.: Elm-art: An intelligent tutoring system on world wide web. In: International conference on intelligent tutoring systems. pp. 261–269. Springer (1996)
4. Bull, S., Kay, J.: Student models that invite the learner in: The smili:() open learner modelling framework. International Journal of Artificial Intelligence in Education **17**(2), 89–120 (2007)
5. Bull, S., Kay, J.: Open learner models as drivers for metacognitive processes. In: Azevedo, R., Aleven, V. (eds.) International Handbook of Metacognition and Learning Technologies, pp. 349–365. Springer (2013)
6. Bull, S., Kay, J.: SMILI: a Framework for Interfaces to Learning Data in Open Learner Models, Learning Analytics and Related Fields. I. J. Artificial Intelligence in Education **26**(1), 293–331 (2016). https://doi.org/10.1007/s40593-015-0090-8, https://doi.org/10.1007/s40593-015-0090-8
7. Cook, R., Kay, J., Kummerfeld, B.: Mooclm: User modelling for moocs. In: User Modeling, Adaptation and Personalization - 23rd International Conference, UMAP 2015, Dublin, Ireland, June 29 - July 3, 2015. Proceedings. pp. 80–91 (2015)
8. Guerra-Hollstein, J., Barria-Pineda, J., Schunn, C.D., Bull, S., Brusilovsky, P.: Fine-grained open learner models: complexity versus support. In: Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization. pp. 41–49. ACM (2017)
9. Hsiao, I.H., Sosnovsky, S., Brusilovsky, P.: Guiding students to the right questions: adaptive navigation support in an e-learning system for java programming. Journal of Computer Assisted Learning **26**(4), 270–283 (2010)
10. Kay, J.: The um toolkit for cooperative user modelling. User Modeling and User-Adapted Interaction **4**(3), 149–196 (1994)
11. Kay, J.: Learner know thyself: Student models to give learner control and responsibility. In: Proceedings of International Conference on Computers in Education. pp. 17–24 (1997)

12. Kay, J., Kummerfeld, B.: Creating personalized systems that people can scrutinize and control: Drivers, principles and experience. ACM Trans. Interact. Intell. Syst. **2**(4), 24:1–24:42 (Jan 2013). https://doi.org/10.1145/2395123.2395129, http://doi.acm.org/10.1145/2395123.2395129
13. Miller, B., Ranum, D.: Runestone interactive: tools for creating interactive course materials. In: Proceedings of the first ACM conference on Learning@ scale conference. pp. 213–214. ACM (2014)
14. Miller, B., Resnick, P., Ericson, B.: Using and customizing open-source runestone ebooks for computer science classes (abstract only). In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. pp. 741–741. SIGCSE '17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/3017680.3017814, http://doi.acm.org/10.1145/3017680.3017814
15. Tang, L.M., Kay, J.: Scaffolding for an olm for long-term physical activity goals. In: Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization. pp. 147–156. UMAP '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3209219.3209220, http://doi.acm.org/10.1145/3209219.3209220