

Towards Distributed Intelligent Tutoring Systems Based on User-owned Progress and Performance Data

Yoshi Malaise*, Maxim Van de Wynckel and Beat Signer

Web & Information Systems Engineering Lab, Vrije Universiteit Brussel, 1050 Brussels, Belgium

Abstract

The use of recommendation engines to personalise students' learning experiences can be beneficial by providing them with exercises tailored to their knowledge. However, the use of these systems often comes at a cost. Most learning or tutoring systems require the data to be stored locally within a proprietary database, limiting learners' freedom as they move across different systems during their learning journey. In addition, these systems might potentially cause additional stress, as the learner might feel observed without knowing who has access to their learning progress and performance data. We propose a solution to this problem by decentralising learning progress and performance data in user-owned Solid Pods. We outline the proposed solution by describing how it might be applied to an existing environment for programming education that already includes research on how to align difficulty levels of exercises across different systems.

Keywords

Personal Data Vault, Intelligent Tutoring Systems, Exercise Recommendation

1. Introduction

The use of Intelligent Tutoring Systems (ITS) to adapt the recommendation of exercises based on individual learners has proven to be effective [1]. Evidence suggests that more personalised learning leads to increased learner agency, self-reliance and motivation [2]. However, the move towards smart learning systems is not without costs. Intelligent Tutoring Systems typically require that all of the exercises are stored in one centralised system that is often also monitored by teachers and persons who are responsible for ultimately grading the students. Previous studies revealed that implementing electronic performance monitoring can result in lower job satisfaction, increased stress levels as well as reduced autonomy, and can further be perceived as a violation of trust [3]. In ideal circumstances, the learning environments should also offer a safe space for students to practise on problems they find interesting or challenging, without having to worry about how they will be perceived by teachers. For example, a student might be concerned that the perception of the teacher might change if they are still shown to be struggling with material from previous years.

In the following, we propose a solution to this problem based on the Solid specification [4] where (a) *the same user progress can be used across multiple learning applications* to provide a

The 1st Solid Symposium Poster Session, co-located with the 2nd Solid Symposium, May 02 – 03, 2024, Leuven, Belgium

✉ ymalaise@vub.be (Y. Malaise); mvdewync@vub.be (M. Van de Wynckel); bsigner@vub.be (B. Signer)

🌐 <https://wise.vub.ac.be/member/yoshi-malaise> (Y. Malaise); <https://maximvdw.be/profile/card> (M. Van de Wynckel); <https://beatsigner.com> (B. Signer)

🆔 0000-0002-3228-6790 (Y. Malaise); 0000-0003-0314-7107 (M. Van de Wynckel); 0000-0001-9916-0837 (B. Signer)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

smooth flow state and (b) *a student can opt in to share progress with individual applications for training the model*, but without having to provide teachers access to their entire performance for individual exercises. We discuss the solution by answering some open questions posed by Malaise and Signer [5], who introduced the *Explorotron* prototype for programming education. In this prototype, there are multiple smaller sub-applications called *study lenses*, each taking a source code file and generating exercises based on certain difficulty levels matching the PRIMM methodology [6]. While their tool offers the possibility to suggest learning experiences based on logical progression through built-in study lenses, the authors conclude with a few major challenges to overcome:

- Students can generate exercises from any source file, so generated content is never used by other students, leading to a cold start problem that makes recommendations challenging.
- Students should be free to decide which data they want to share and with whom they want to share it.
- Third-party developers should be able to contribute custom exercise generators (study lenses) transparently—the same recommendations/profiling should work across the board.

2. Solution

For our demonstration, we assume that student modelling happens through the use of Bayesian Knowledge Tracing (BKT) [7], a common approach in the design of Intelligent Tutoring Systems. However, note that with some modifications, other approaches could also be supported. BKT is a probabilistic model that requires four parameters to be fitted for each learning object as illustrated in Figure 1.

- $P(L_t)$: The probability that the topic being covered by the learning object is mastered at time t . A value $P(L_0)$ has to be provided to indicate the chance the user knows the topic before attempting any exercise. Note that the topics are defined as a combination of the knowledge topic and the difficulty level as defined later in Section 2.1.
- $P(S)$: The probability that the user makes a *slip*, i.e. gets it wrong even though they know the topic.
- $P(G)$: The probability that the user makes a lucky guess and gets the answer right even though they do not know the topic.
- $P(T)$: The probability that the user learns the topic while performing this exercise.

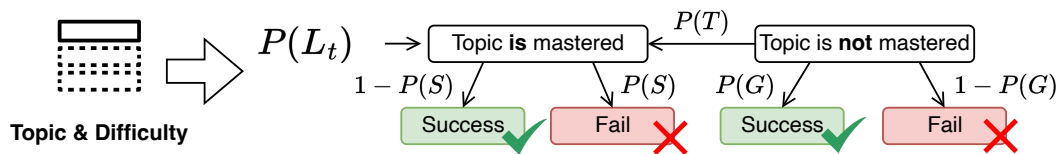


Figure 1: Example of the Bayesian Knowledge Tracing model per topic and difficulty pair

Based on these values, we can then calculate the updated probability that a topic (e.g. “Arrays”) is mastered each time the user is presented with a new exercise about this topic (L_{t+1}) based on whether they succeeded or failed to solve the exercise. It is important to note that all these probabilities are exercise dependent and will, as such, not be shared across the applications except for $P(L_t)$. This value is the probability that the user knows a topic, which can be shared across any application as long as they are referring to the same topic.

We aim to store the result of every exercise the user performs within a personal data vault, together with the calculated $P(L_t)$ for every topic and difficulty level pair as shown in Figure 2. Developers of educational applications could then request limited-time access to all of their users’ entries in order to fit their models without the need to own all the data permanently. Shared information, such as the $P(L_t)$, will make it easier for other developers to contribute their own extensions that provide exercise recommendations.

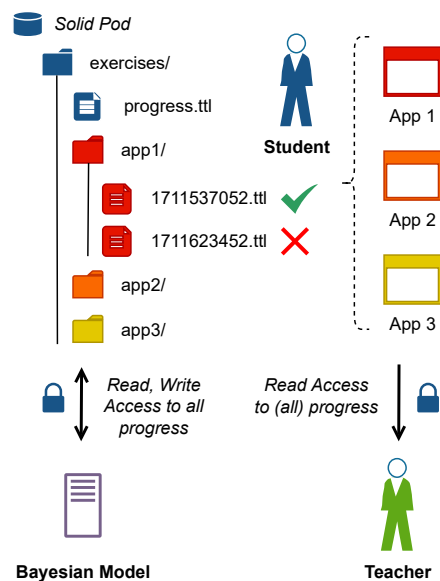


Figure 2: Overview of individual apps storing exercise results in a Solid Pod

```
<> a schema:ReplyAction; # A reply to a question
dct:created "2024-03-28T11:57"^^xsd:dateTime;
schema:agent <http://.../profile/card#me>;
# Exercise question
schema:parentItem [ a schema:Question;
schema:name "What is the output of ...?"@en;
schema:educationalLevel primm:Predict;
schema:eduQuestionType "Flashcard"@en;
foaf:topic dbr:Array;
foaf:agent <https://app1.org/>. # Creator
];
schema:result [ a schema:Answer; # User answer
schema:answerExplanation [ ... ];
schema:review [ a schema:Review;
# The grade of the answer (e.g. 5 out of 5)
schema:reviewRating 5; schema:bestRating 5.
]
].
```

Listing 1: Example reply on a question generated by app #1 using prefixes dbr [8], schema [9], foaf [10] and dct [11]

Users can already indicate their topics of interest in their Solid Pod using the topics of interest predicate from the foaf vocabulary [10], which can aid educational recommendation systems to provide exercises that interest the user. Together with data about the past performance, we can create personalised learning paths for users based on their strengths and weaknesses from multiple sources without creating any vendor lock-in and without claiming ownership of the students’ data.

In addition, the progress of these exercises and topics can optionally be shared with an educator or mentor as illustrated in Figure 2, allowing for personalised feedback and guidance to further support a user’s learning journey. The fine-grained control over the data also enables students to share the information on topics they are currently covering in class, while preventing the need to share that they are still working on other topics they do not want to share.

2.1. Exercises and Progress

Each topic has a set of exercises with their individual difficulty level. Our proposed solution uses the existing PRIMM principles [6] to identify the five stages of difficulties for exercises. These stages include the ability to *predict* the output of a piece of code, *run* a piece of code, *investigate* a piece of code and answer some questions about its structure, *modify* a piece of code so the behaviour changes to a new desired result and finally, *make* a similar piece of code from scratch. Together, these five principles identify the logical progression of difficulty levels for looking at a piece of code from a student's perspective. Listing 1 illustrates an example exercise and the answer given to this exercise that includes the difficulty level¹ using the `educationalLevel` predicate.

A further extension of the architecture would allow third-party developers to share their plugins' capabilities regarding what type of content they could provide. Thus, the recommendation engine can be a fully separate component querying all content providers included by the user without being centrally controlled by one individual.

```
<#progress_arrays_predict> a sosa:ObservableProperty ;
  rdfs:label "Prediction progress"@en ; ssn:isPropertyOf <#me> ;
  foaf:topic dbr:Array ; schema:educationalLevel primm:Predict .
<#1711623452> a sosa:Observation ; #  $P(L_{t+1})$ 
  sosa:usedProcedure dbr:Bayesian_Knowledge_Tracing ;
  dct:created "2024-03-28T11:57"^^xsd:dateTime ;
  foaf:agent <https://app1.org/> ; sosa:observedProperty <#progress_array> ;
  sosa:hasResult [ qudt:floatPercentage "38.12"^^xsd:float ] .
<#1711537052> a sosa:Observation ; #  $P(L_t)$ 
  sosa:usedProcedure dbr:Bayesian_Knowledge_Tracing ;
  dcterms:created "2024-03-27T11:57"^^xsd:dateTime ;
  foaf:agent <https://app1.org/> ; sosa:observedProperty <#progress_array> ;
  sosa:hasResult [ qudt:floatPercentage "25.0"^^xsd:float ] .
```

Listing 2: Multiple observations of user progress for a particular topic and difficulty level

Listing 2 shows multiple observations of user progress for a particular topic at a given time (i.e. $P(L_t)$). Each observation, described using the SOSA ontology [12, 13], is created by an application. This progress is shared, allowing each educational application to use these observations to determine the current knowledge on a topic.

3. Conclusions

In this paper, we analysed a use case of a digital learning platform that helps students learn programming. However, there are currently some open challenges on how personalisation and interoperability with third-party developers might be achieved without taking away the learner's data ownership. We then proposed a novel solution to support those tasks by allowing users to store all their exercise results in Solid Pods and giving them full control over how and

¹The `primm` vocabulary represents a trivial vocabulary to describe the difficulty levels in programming exercises

when applications can access this data. In this work, we assumed that both, users and application developers could be fully trusted. If more validation and authority is needed, additional layers can be built on top of this system, for example, by using a blockchain to verify claims, as described in [14, 15].

References

- [1] J. A. Kulik, J. D. Fletcher, Effectiveness of Intelligent Tutoring Systems: A Meta-analytic Review, *Review of Educational Research* 86 (2016). doi:10.3102/0034654315581420.
- [2] V. Prain, P. Cox, C. Deed, J. Dorman, D. Edwards, C. Farrelly, M. Keeffe, V. Lovejoy, L. Mow, P. Sellings, B. Waldrup, Z. Yager, Personalised Learning: Lessons to be Learnt, *British Educational Research Journal* (2013). doi:10.1080/01411926.2012.669747.
- [3] R. Siegel, C. J. König, V. Lazar, The Impact of Electronic Monitoring on Employees' Job Satisfaction, Stress, Performance, and Counterproductive Work Behavior: A Meta-Analysis, *Computers in Human Behavior Reports* 8 (2022). doi:10.1016/j.chbr.2022.100227.
- [4] S. Capadisli, T. Berners-Lee, R. Verborgh, K. Kjernsmo, Solid Protocol, 2022. <https://solidproject.org/TR/protocol>.
- [5] Y. Malaise, B. Signer, Explorotron: An IDE Extension for Guided and Independent Code Exploration and Learning (Discussion Paper), in: *Proceedings of Koli Calling 2023, 23rd International Conference on Computing Education Research, Koli, Finland, 2023*. doi:10.1145/3631802.3631816.
- [6] S. Sentance, J. Waite, M. Kallia, Teaching Computer Programming with PRIMM: A Sociocultural Perspective, *Computer Science Education* 29 (2019). doi:10.1080/08993408.2019.1608781.
- [7] O. Bulut, J. Shin, S. N. Yildirim-Erbasli, G. Gorgun, Z. A. Pardos, An Introduction to Bayesian Knowledge Tracing With PyBKT, *Psych* 5 (2023). doi:10.3390/psych5030050.
- [8] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, DBpedia: A Nucleus for a Web of Open Data, in: *Proceedings of ISWC 2007, 6th International Semantic Web Conference, Busan Korea, 2007*. doi:10.1007/978-3-540-76298-0_52.
- [9] A. Iliadis, A. Acker, W. Stevens, S. B. Kavakli, One Schema to Rule Them All: How Schema.org Models the World of Search, *Journal of the Association for Information Science and Technology* (2023). doi:doi.org/10.1002/asi.24744.
- [10] D. Brickley, L. Miller, FOAF Vocabulary Specification 0.99, 2014. <http://xmlns.com/foaf/spec/>.
- [11] S. Weibel, J. Kunze, C. Lagoze, M. Wolf, RFC2413: Dublin Core Metadata for Resource Discovery, Technical Report, 1998. doi:10.17487/RFC2413.
- [12] K. Janowicz, M. Compton, The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration Into the Semantic Sensor Network Ontology, in: *Proceedings of SSN 2010, 3rd International Conference on Semantic Sensor Networks, Shanghai, China, 2010*. <https://ceur-ws.org/Vol-668/paper12.pdf>.
- [13] K. Janowicz, A. Haller, S. J. Cox, D. Le Phuoc, M. Lefrançois, SOSA: A Lightweight Ontology for Sensors, Observations, Samples, and Actuators, *Journal of Web Semantics* 56 (2019). doi:10.1016/j.websem.2018.06.003.

- [14] N. Chowdhury, M. Ramachandran, A. Third, A. Mikroyannidis, M. Bachler, J. Domingue, Towards a Blockchain-based Decentralised Educational Landscape, in: Proceedings of eLmL 2020, 12th International Conference on Mobile, Hybrid, and On-line Learning, Valencia, Spain, 2020. <https://oro.open.ac.uk/69608/>.
- [15] A. Mikroyannidis, Blockchain Applications in Education: A Case Study in Lifelong Learning, in: Proceedings of eLmL 2020, 12th International Conference on Mobile, Hybrid, and On-line Learning, Valencia, Spain, 2020. <https://oro.open.ac.uk/69593/>.