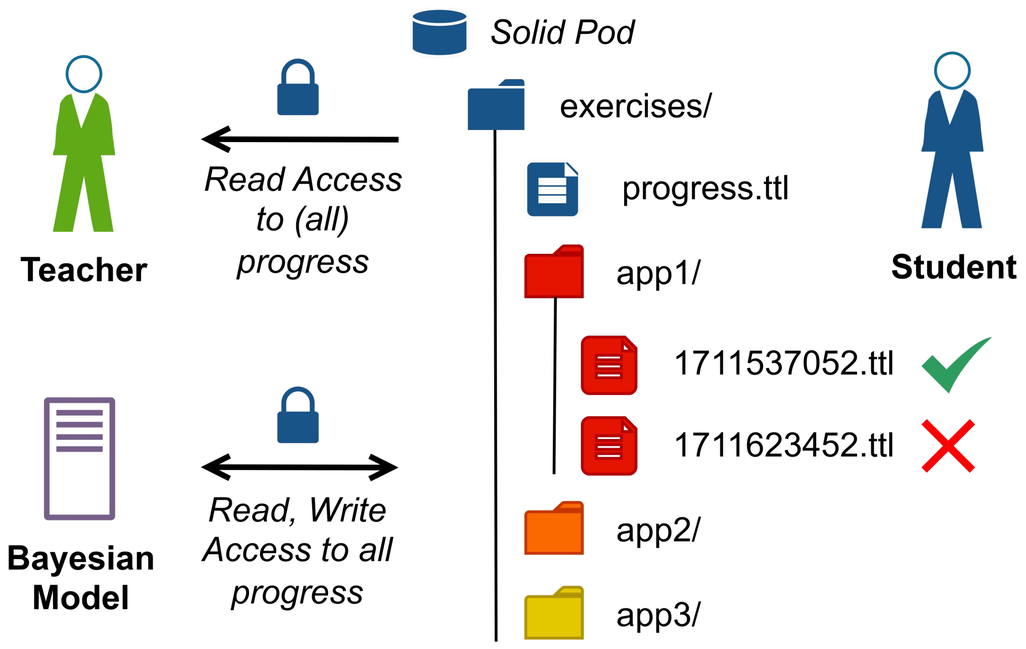
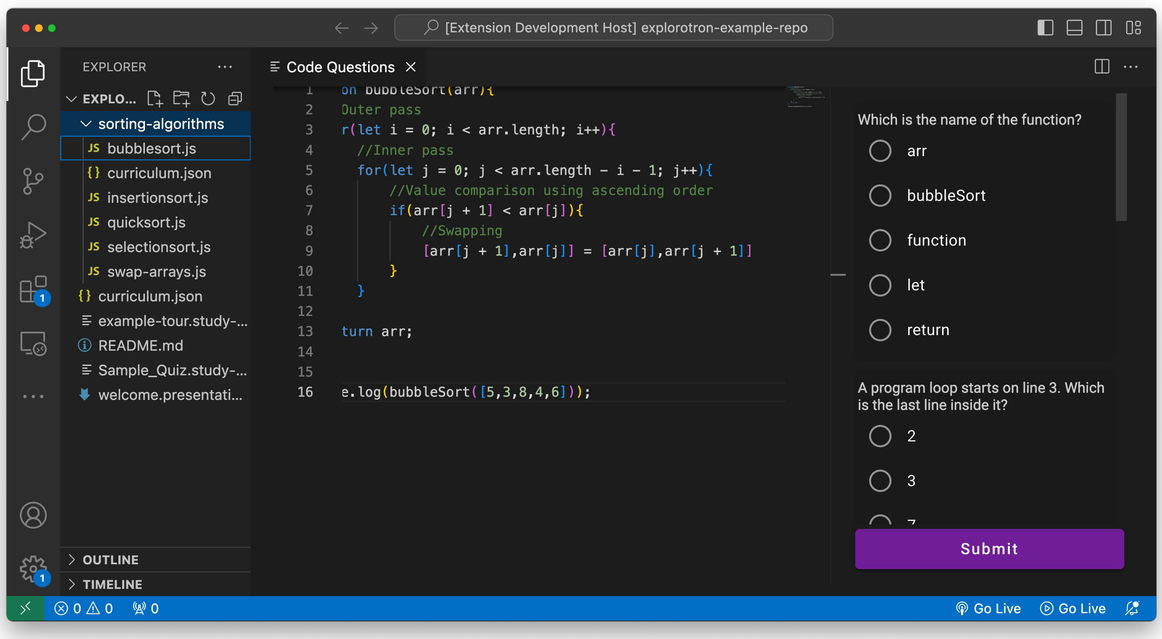
Towards Distributed Intelligent Tutoring Systems Based on User-owned Progress and Performance Data

VRIJE UNIVERSITEIT BRUSSEL

The use of recommendation engines to personalise students' learning experiences can be beneficial by providing them with exercises that are 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 the freedom of the learner 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.

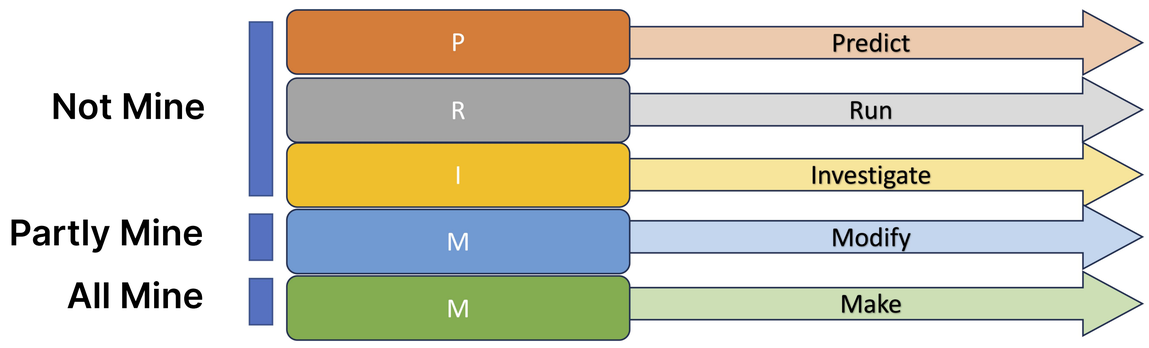


INTRODUCTION



Example of a tutoring system for programming exercises

As part of an overarching project we have developed the Explorotron Visual Studio Code extension that is designed to help students **learn** from arbitrary JavaScript code examples by providing different interactive views or so-called **study lenses** each focusing on different aspects of the code. The individual study lenses aim to generate **user-tailored exercises** that are suitable to be used the different stages of the **PRIMM method**, which offers an objective way to indicate the difficulty level of an exercise.



PRIMM principles used to indicate the difficulty level of programming exercises

In order to provide user-tailored exercises that make use of the complete progress of a student over multiple applications and tutoring systems, we propose a **decentralised storage for learning progress** of individual topics using Solid. Students can control the **access rights** of individual applications and other users (e.g. teachers) to their learning progress.

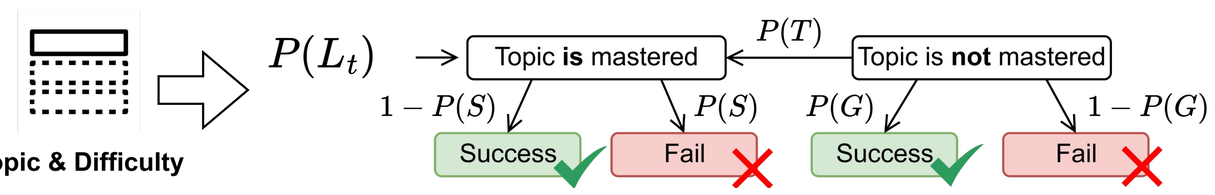
BAYESIAN KNOWLEDGE TRACING AS A METHOD TO DETERMINE PROGRESS

$P(L_t)$ The probability that the topic being covered by the topic and difficulty pair is mastered at a given time t. A value $P(L_0)$ has to be provided to indicate the chance the user knows the topic before attempting any exercise.

$P(S)$ The probability that the user gets it wrong even though they know the topic.

$P(G)$ The probability that the user gets the answer right even though they do not know the topic.

$P(T)$ The probability that the user actually learns the topic while performing an exercise.



PROPOSED SOLUTION

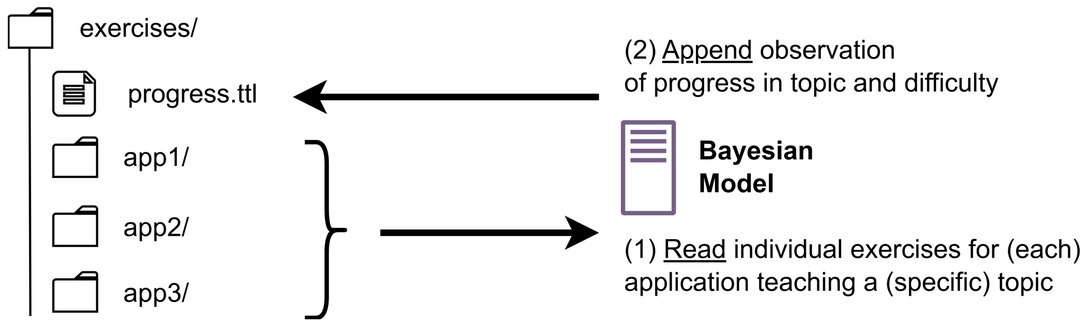
```
<> a schema:ReplyAction; # A reply to a question
  dct:created "2024-03-28T11:57"^^xsd:dateTime ;
  schema:agent <http://.../profile/card#me> ;
  # Exercise question
  schema:parentItem [ a schema:Question ;
    schema:name "What is the output of ...?"@en ;
    schema:educationalLevel primm:Predict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

Example of an answer to an exercise question for the topic "Array" and difficulty level indicated using the PRIMM principles (the primm vocabulary is currently not public but indicates the existing PRIMM principles)

Individual questions are defined using a **schema.org** "Question". Each question has a difficulty level, topic and the agent that generated or asked the question. An "Answer" is provided with a rating that indicates if the student succeeded to provide a valid answer.

Each answer is added to the Solid Pod using a **schema:ReplyAction** for each individual learning platform or application.

A Bayesian model will have read access to individual exercises (1) of the user in order to append a new **observation** to the progress of $P(L_t)$ (2).




```
<#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 ] .
```

Example of progress using Bayesian Knowledge Tracing per topic and difficulty level

Yoshi Malaise
ymalaise@vub.be
0000-0002-3228-6790

Maxim Van de Wynckel
mvdewync@vub.be
0000-0003-0314-7107

Beat Signer
bsigner@vub.be
0000-0001-9916-0837