



**QUEEN'S
UNIVERSITY
BELFAST**

Creating a Dialogue-Based Tutorial in Questionmark

O'Neill, I. (2017). Creating a Dialogue-Based Tutorial in Questionmark. *Reflections*, June 2017, 8.

Published in:
Reflections

Document Version:
Publisher's PDF, also known as Version of record

Queen's University Belfast - Research Portal:
[Link to publication record in Queen's University Belfast Research Portal](#)

Publisher rights

© 2017 QUB & The Author.

This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

General rights

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Creating a Dialogue-Based Tutorial in Questionmark

By Dr Ian O'Neill, Electronics, Electrical Engineering and Computer Science

“Very large numbers of students on key pathways,” “restructured Academic Year,” “more year-long modules,” - just a few of the developments at Queen’s that encouraged this lecturer to see if assessment technology might be used inventively to enhance the student (and lecturer) experience in a challenging academic environment.

My aim was to re-create in software, in a simplified form, the kind of person-to-person, question-and-answer exchanges that characterise tutorials in small groups. In this case, the tutorial would be implemented in Questionmark, an assessment package that is familiar to staff and students at Queen’s, and a proven means of evaluating work and providing feedback.

With its bias towards structured, text-based interaction, Questionmark leads to a tightly constrained dialogue, but dialogue nonetheless. It enables tutors to compose, with relative ease, sets of written questions and possible answers. The tutor can also build feedback into the system, comments that are displayed whenever the student chooses an answer, or a combination of answers.

Questionmark can make especially good use of the pattern of ticks that result from an open-ended ‘multiple response question’ (“What can you tell me about this diagram?”), where the student is allowed to select freely from a generous selection of correct and incorrect answers. Just as an experienced tutor would do, the system responds to the pattern, offering tips on what went well – or more importantly, on what went wrong and why, when the pattern of answers is indicative of a common misunderstanding of a concept, technique or terminology. The system can also pose a follow-up question that gives the student the opportunity to try again, or that focusses just on the area of weakness. Of course, all this presupposes that the control logic

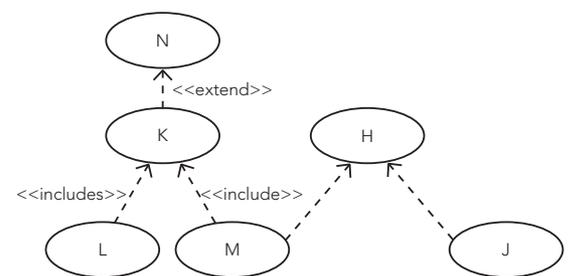
of the tutorial is correctly authored and tested, so that the system mimics the behaviour of the experienced tutor: in Questionmark terms, each tutorial is an ‘assessment’ with subtle, conditional ‘Jump Blocks’ that take the student, under a particular set of circumstances, to relevant follow-up ‘Question Blocks’.

In creating my Questionmark tutorial, I set out to convey the sort of constructive feedback that I would give Level 2 Software Engineering students when they are learning to interpret Use Case Diagrams, a form of UML¹ notation that is used to represent important relationships between sets of requirements for a computer system. Use Case Diagrams require only a small set of symbols. However, having taught and found value in the Use Case approach for several years, I am aware that Use Case Diagrams pose difficulties, some of which arise from the specialised terminology associated with the diagrams, and some from the quite subtle messages conveyed by the different arrow styles and labels in the diagrams themselves.

With the benefit of experience, the tutor can predict and have the system react to such problems when they occur. For example, in the Use Case tutorial, when the students’ incorrect answers resulted from a misinterpretation of the symbols of the UML, Questionmark gave them feedback on their overall answer combination, then on each of their answers individually, before asking

them to choose from the symbol-related answers again. Once the narrow, symbol-related answers had been mastered, the system would return to the broader question: “What can you tell me about this diagram?”

‘Automated tutorials’ like this have the advantage that they give students an opportunity to revisit topics whenever it suits them, particularly during those year-long modules. Students can test their grasp of concepts as they are introduced, or during their preparation for formal assessment many months later. It is more than convenient self-service. In the context of large and growing class sizes, a suitable means of automating some of the more routine student-tutor interactions is of benefit to tutors and students alike, freeing tutors’ and students’ time for face-to-face discussion of more challenging topics, and probably more interesting ones.



A simple use case diagram

The first reactions to the Use Case tutorial were certainly very favourable. As one student on the Software Engineering module put it: “Brilliant idea – wish there was one [tutorial] for each topic.” So, a good start, and a challenge to provide more! As technologies for learning evolve – exploiting text, speech, and indeed multiple modes of interaction – we can expect to find many more occasions when computers can usefully play their part as virtual tutors.

Please contact i.oneill@qub.ac.uk for further details.

¹ See: Booch, Rumbaugh and Jacobson *The Unified Modeling Language [UML] User Guide, 2nd Edition*, Pearson, 2005; and Dennis, Wixom and Tegarden, *Systems Analysis & Design – An Object-Oriented Approach with UML, 5th Edition*, Wiley 2015.