

Question Generation and Adaptation Using a Bayesian Network of the Learner's Achievements

Michael Wißner¹, Floris Linnebank², Jochem Liem²,
Bert Bredeweg², and Elisabeth André¹

¹ Human Centered Multimedia, Augsburg University, Germany
{wissner, andre}@informatik.uni-augsburg.de

² Informatics Institute, University of Amsterdam, The Netherlands
{f.e.linnebank, j.liem, b.bredeweg}@uva.nl

Abstract. This paper presents a domain independent question generation and interaction procedure that automatically generates multiple-choice questions for conceptual models created with Qualitative Reasoning vocabulary. A Bayesian Network is deployed that captures the learning progress based on the answers provided by the learner. The likelihood of concepts being known or unknown on behalf of the learner determines the focus, and the question generator adjusts the contents of its questions accordingly. As a use case, the Quiz mode is introduced.

1 Introduction

The DynaLearn project (<http://www.DynaLearn.eu>) has developed an Interactive Learning Environment (ILE) that supports learners in manipulating conceptual knowledge using Qualitative Reasoning technology [1]. Learners learn by creating conceptual models using the software (an example is shown in Figure 1). However, the ILE can also be used to have learners learn from interacting with an existing model (e.g. made by a teacher, domain expert or maybe a peer). One of the instruments developed for this is the *Quiz*-mode, essentially a question/answering interaction that engages a learner to discover the knowledge captured in an already existing model.

In DynaLearn, specific constraints apply: The question generation has to proceed automatically, be domain independent and adapt to the ongoing learning on behalf of the learner. To illustrate the components and ideas presented, consider the architecture and interaction flow for the Quiz-mode as shown in Figure 2. Everything starts with the learners working on a model, which they *Build* and *Simulate*. Then the Quiz option can be activated and the *Bayesian Network* (BN) for knowledge tracing is built from the *QR model & simulation* (dashed line) (Section 2.1). Next, the *Generate questions* component constructs a *Question list* (Section 2.2). The *Question request* influences this process and as it

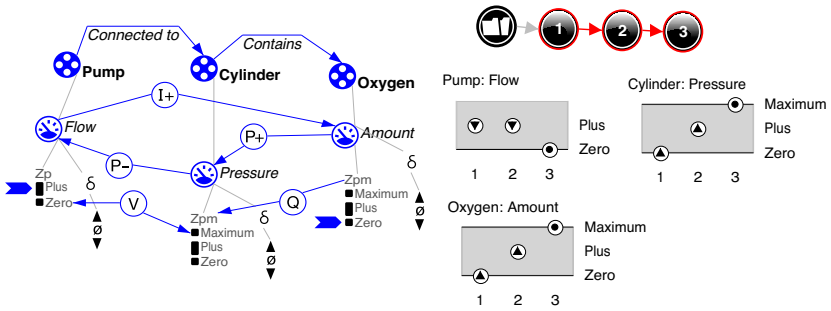


Fig. 1. An example model describing oxygen flow (LHS) along with its simulation results: State-graph and value histories (RHS)

contains those concepts that are least understood by the learner. The *Select question* takes a specific *Question* from the list based on its type (preferring the least asked question types for variety) and difficulty (trying to present a gentle learning curve to the learner). The *Dialogue history* keeps record of all asked questions and can thus support the selection task and prevent obvious repetition. The selected question is then presented to the learner in the form of a multiple-choice question (*Express question*). *Check answer* assesses the answer given by the learner. An answer is either correct or incorrect, and with that information the *Bayesian Network* is updated as well as the *Dialogue history*. The new state of the BN is analyzed by *Determine focus* to establish the next *Question request*, again based on the now least known concept. The latter feeds into *Generate questions* to steer the next round of the Quiz mode. Each Quiz lasts for a number of interaction rounds until either the system determines that the learner’s knowledge of the current model is sufficient, or there are no more questions to ask.

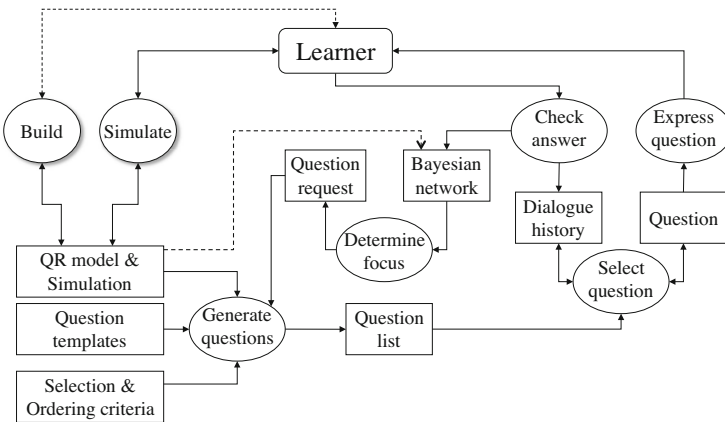


Fig. 2. System Architecture and Data Flow

2 Implementation

2.1 Question Generation

The question generator is based on QUAGS [2]. It is implemented addressing two requirements. First, the component is domain independent. It uses generic ingredient types as the basis to generate questions. These generic types are instantiated with domain specific concepts and therefore the questions are in fact about the domain itself. Second, the number of questions that can be generated for a specific conceptual model and its simulation is enormous. For instance, the Cerrado succession model [3] easily allows for over 72000 questions to be generated. Many of these questions are rather simple and do not necessarily address the key concepts in the model (for instance, asking for the value of each quantity in each state). The generator has two features to ensure that relevant questions are generated and that the total number of questions stays small. One concerns the notion of a focus (Question Request in Figure 2). The generator accepts a focus and adjusts the generation of questions accordingly. In addition, the generator has intrinsic mechanisms that limit the number of questions it will generate ($N < 16$), and that ensures that those generated questions address the most relevant domain facts in the simulation. So, even in the absence of a focus, a limited set of relevant questions will be generated. Finally, QUAGS is augmented to generate multiple choice questions.

2.2 Knowledge Tracing with the BN

From the conceptual model a BN is created to track the learner's performance. Following Millán et al. [4], entities can be regarded as the subject nodes, quantities as the topic nodes and every ingredient associated with a quantity can be regarded as a concept node. That is, for learners to understand an entity, they must know all of its quantities, and to know a quantity, they must understand all the concepts *directly related to it*.

To represent the learner's knowledge of these different concepts, each concept node in our BN can be either *known* or *unknown*. Thus our approach follows ideas as presented by Corbett and Anderson [5]. While their model does not implement *forgetting*, it does take the possibility of *guesses* (a learner does not know the answer but answers correctly) and *slips* (a learner answering wrong despite knowing the answer) into consideration. To represent this in our model, each concept node is given a child node representing one question, and the possibility of guesses and slips is modeled in this question node. Finally, all entity nodes are connected to the node *Model*, which represents the learner's knowledge about the model as a whole and is used to determine whether a model is sufficiently understood by the learner.

However, there is one problem: Imagine an extension of the model shown in Figure 1, but with two containers (a cylinder and a scuba tank), connected by a hose, and the cylinder being filled with oxygen from a pump. This brings up the issue of *recurring concepts*: If a student learns something about one container,

they will have learned something about the other as well. Therefore, the idea of *answer collectors* is introduced, which adds an additional layer to the network (between the concept nodes and their question nodes). In case of a concept with multiple instances, each of them will have an answer collector which propagates the evidence from the question node to all the instances. This way, if learners answer a question about one of the instances, their knowledge of all the instances will increase, although the answer collectors will have more impact on “their” instances than on the others.

3 Conclusion

Being able to ask relevant questions is an important requirement for an ILE. We have presented a set of components for creating this functionality within the DynaLearn ILE. The question generation functionality is domain independent. It has a mechanism to ensure that questions are meaningful, and that the total number of questions stays within limits. A BN is used to track the learner’s performance and steer the question generator.

Further improvements could address alternative methods for updating the learner’s knowledge: Multiple question nodes could be connected to a concept and their exact number would be dependent from the type of concept (more questions for more important concepts). Also each concept currently makes an equal contribution to its quantity’s understanding. The approach could be extended to allow for adding weights to concept nodes in the BN, e.g. making magnitudes more important than derivatives. Future research could also address the re-usage of the BN across multiple models of the same domain, since a single QR model can be a part of a bigger domain a learner is supposed to cover. Our BN setup has the basics to incorporate these extensions.

Acknowledgments. This research is co-funded by EC FP7, Project no. 231526, <http://www.DynaLearn.eu>. Our implementation uses SMILE and GEeNIe from the Decision Systems Laboratory, Univ. of Pittsburgh, <http://genie.sis.pitt.edu/>.

References

1. Bredeweg, B., Linnebank, F., Bouwer, A., Liem, J.: Garp3 – workbench for qualitative modelling and simulation. *Ecological Informatics* 4, 263–281 (2009); Special Issue: Qualitative models of ecological systems
2. Goddijn, F., Bouwer, A., Bredeweg, B.: Automatically generating tutoring questions for qualitative simulations. In: *Proc. of the 17th Int. Workshop on Qualitative Reasoning*, pp. 87–94 (2003)
3. Salles, P., Bredeweg, B.: Modelling population and community dynamics with qualitative reasoning. *Ecological Modelling* 195, 114–128 (2006)
4. Millán, E., Pérez-de-la-Cruz, J.L., Suarez, E.: Adaptive bayesian networks for multi-level student modelling. In: Gauthier, G., VanLehn, K., Frasson, C. (eds.) *ITS 2000. LNCS*, vol. 1839, pp. 534–543. Springer, Heidelberg (2000)
5. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMUAI* 4, 253–278 (1995)