INNOVATION AWARD

Galore: A Platform for Experiential Learning

Abhishek Parakh
Cybersecurity
University of Nebraska at Omaha
Omaha, NE
aparakh@unomaha.edu

Mahadevan Subramaniam
Computer Science
University of Nebraska at Omaha
Omaha, NE
msubramaniam@unomaha.edu

Abstract—The use of customizable learning objects in multiple different formats such as visual, auditory, text, interactive widgets and newly defined learning objects called gamelets have a potential to tremendously enhance experiential learning. A parameterized environment, called Galore, that integrates such learning objects into a seamless experience based on student learning styles and preferences for teaching difficult counter intuitive concepts in quantum communications is described.

Keywords—Learning objects, gamelets, experiential learning, quantum computing, quantum cryptography

I. Introduction

In recent years, digital learning environments enhanced by the integration of serious games and learning objects has shown tremendous success in several domains such as management, medicine, STEM disciplines as well as information technology. Learning object repositories (LOR), such as Clark [1], although a useful resource, typically focus on illuminating core concepts using various forms of education aides such as videos, textual, interactive and non-interactive widgets, codingbats, as well as assessments based on exercises, quizzes and tests as opposed to student engagement. On the other hand, serious games excel at enhancing engagement through interactions, active experimentation and explorations, missions as well as competition and create a state of flow amongst the players.

Both forms of educational tools, LORs and serious games provide some level of customization based on student learning styles and preferences and can statically or dynamically generate new scenarios, learning pathways, and synthesize new exercises and validate dependencies. However, assimilating intentional user actions and deliberate interactions in a game-based learning environment has the tendency to break the flow of the game. In this paper, we propose integrating LORs and serious games into a single learning platform to enhance and employ the benefits of both the educational tools. Such an integration can help ease the adoption of these platforms into a classroom environment provide students with an in-depth, engaging and holistic learning experience for students.

The new learning platform, Galore, proposed in this paper is built upon the four phases of Kolb's experiential learning cycle. The concrete and active experiential phases of the learning cycle are implemented using extended LORs that incorporate gamelets. These LORs and gamelets are then synthesized into scenarios, exercises and interactions that provide opportunities for reflection and abstract conceptualizations conforming to Kolb's learning cycle. Object in the LOR, example Clark, and gamelets are grouped into distinct Units and further subdivided into nano-modules, micro-modules, and modules based on the expected completion times.

All of the components of the Units and gamelets are parameterized with metadata that allows students to consume these in their preferred learning style and learning outcomes. Some of the parameters of these components include prerequisites, learning attributes such as examples, explanations, definitions, interactivity level, hint levels, etc. as well as various learning outcomes supported by that component. Based on the desired learning objectives and set of parameters as input from the student, Galore can automatically synthesize several lesson plans for students to review. A student can then explore these lesson plans and choose one based on their performance and learning predisposition. The synthesized lesson plans ensure that all the learning outcome dependencies are satisfied and a lesson is considered completed by the student when the student is able to demonstrate their proficiency in all the learning objects associated with a lesson.

Recently, Jupyter notebooks have become the defacto standard for building educational tools in quantum computing [2] and consequently we adopt them for our proposed platform as well. The synthesized lesson plans are presented to a student using Jupyter notebooks with a cell (or a collection of cells) corresponding to different learning objects. The various cells in the notebook and collection of notebooks are linked together via three graphs. Two of them enforce concept and outcome dependencies while the third

T A. Parakh wishes to acknowledge that the research summarized in this article is partly supported by a grant from University of Nebraska at Omaha under the U.S. Department of State-supported initiative Partnership 2020: US- India Higher Education Cooperation.

graph depicts the various learning pathways that a student can take through the lesson plan. Each link in the third graph allows students to interactively traverse from one learning object to the other. The existence of these links, however, is predicated on meeting concept and outcome dependencies. The pathways are initially constructed through student inputs such as learning goals/objectives and preferences. However, as the student traverses between these learning objects, some of the pathways are customized on the fly based on student performance. The platform can automatically generate new lesson plans for future interactions or choose a different lesson plan that is more likely to enable the student to achieve the learning goals and objectives.

The framework proposed in this paper is applied to concepts and learning objectives in the field of secure quantum communication which in turn requires knowing quantum computing and quantum cryptography. Secure quantum internet is undoubtedly the next frontier in the evolution of secure communication. A large scale effort to establish nationwide secure quantum internet is currently underway through the National Quantum Initiative Act of 2018 (quantum.gov). The development of a workforce that can design and build protocols for such a quantum internet needs to be well versed in diverse fields of classical and quantum computing principles, classical and quantum information theory, secure quantum key exchange protocols, networking as well as have an understanding of Physics and optics. Learning and teaching students such a diverse set of topics within existing computer science or cybersecurity curriculum is quite challenging given that textbooks that cover such diverse topics are nearly nonexistent. The proposed platform bridges this gap through the development of parameterized learning objects and gamelets that model these difficult concepts in small digestible chunks.

II. Related Work

Serious games: Game-based learning, and game-based teaching have a deep and rich history in education and training, especially in cybersecurity. The use of games for education has a rich and deep history [3]–[6]. Using text-based multi-player games like MOO and multi-user dungeons (MUDs) in formal educational settings have also been explored earlier [6], [7] and are enjoying a new resurgence with the examinations on the integration of serious games in several learning contexts such as classrooms, online classes, projects and assignments [8]–[12].

In [8], Nunes and McPherson suggest learning environment designers to consider pedagogical learning models in order to incorporate features of the learning model. They argue that designers must take pedagogy into account in order for a learning environment to appropriately reflect the type of learning and the process of learning that will take place in that environment. The work in [12], de Freitas et al is another step in this direction. They investigate several empirical studies involving serious games. They conclude that in order to realize the full potential of game-based learning, a tighter and fine-grained integration of pedagogical approaches employed in formal educational contexts with those underlying games, facilitated by various instructional roles is needed. The work by Thorkild [11] is similar in spirit, but instead focuses on a methodology to build a correspondence between game scenarios and the traditional curricular and pedagogical practices. Their approach emphasizes game-based teaching and describes how teachers can move between game-based and traditional lecture-based modes to incorporate games into traditional instruction. They define multiple roles for teachers such as the instructor, evaluator, playmaker, and guide, to engage with students in the game scenarios. The work by Ketamo et al [10] emphasizes the role of teachers and investigates teacher models in addition to student and learning models to assess the effectiveness of serious games. They consider several games such as the AnimalClass, Eedu Elements, Media Detective, and ALICE FireEvacuation with different teacher models including those supported by flipped classrooms, learners as teachers and teachers as active rivals in serious games. They show that incorporating teacher models into serious game design and scenarios leads to significant improvement in learning outcomes.

E-learning: E-learning approaches using learning objects and associated learning repositories constitute an alternate approach for organizing knowledge concepts into digital learning assets and have been extensively studied as well. Interdisciplinary learning object repositories as well as those focused on an individual discipline have been developed as a resource for students and teachers. A notable learning object repository for cybersecurity [1] has been recently developed incorporating learning digital assets spanning several KUs in different formats. The LOR [1] is a rich source of cybersecurity concepts to students and teachers and cross-indexes the associated learning outcomes to those provided by several certifying agencies. However, like most available LORs, it is a compendium of learning objects and provides limited support in-terms of findability, accessibility, interoperability, and reuse of the learning objects. Most importantly, most of these repositories hardly provide any support to automatically generate learning content for a particular learning experience such as lecture, workshop, tutorial, or a course. Often, instructors have to search/browse LORs to manually collate the materials and organize them into a presentation, lecture notes or materials for a course based on associated outcome and concept dependencies.

Adaptive Rule-based Frameworks: Adaptive rule-based frameworks lesson plan generation have been extensively investigated by several earlier works to automatically synthesize learning materials [13], [14] for a given group of KUs, SLOs, and pedagogical models. Large LORs parameterized by concept and student backgrounds have been developed for CS education earlier [1], [15]–[17]. These works often employ the notion of generative learning objects representing related learning object instances that can be automatically generated on demand by instantiating the generative object. We are unaware of rule-based generative frameworks for cybersecurity that support automatic lesson plan generation. Besides, the rule-based adaptive frameworks, in general, do not pay adequate attention to incorporating and adapting serious games.

Our work in this paper, develops a comprehensive learning environment that integrates serious games, e-learning based LORs and adaptive rule-based frameworks to synthesize materials in a push-button manner for cybersecurity education and training. We focus on the concepts related to quantum cryptography in this work. We use gamelets and adaptive, generative learning objects to seamlessly integrate these three themes. In addition, our environment *Galore* includes components that explicitly realize the four-phases of the experiential learning model into the LORs and gamelets. To the best of our knowledge *Galore* is the first such environment that has been developed for cybersecurity focused on quantum cryptography concepts.

III. GALORE ARCHITECTURE

A student starts Galore by logging into and entering a profile consisting of their background knowledge, and learning preferences. Galore consists of three layers. The bottom most layer is made up of various learning objects such as text, visual, video, interactive and non-interactive widgets and gamelets. The reflection and experimentation zones within this bottom layer provides students with a place for abstract conceptualization. The Galore engine residing in the middle layer is responsible for retrieval, synthesis and instrumentation of the learning objects based on user profile. And finally, the top layer consists of the User Interface based on Jupyter notebooks. The top layer and the middle layers are connected together through a data driven user adaptation engine. The adaptation engine monitors user performance through interactions and when possible either updates the Jupyter notebooks on the fly by dynamically presenting various cells in the notebook or by changing the lesson plan and synthesizing new Jupyter notebooks to aid the student in achieving their desired learning outcomes.

A. *Learning Objectives*

On the surface, learning objects can be thought of as various sections in a traditional textbook that covers a concept. These sections, often, assume a set of pre-requisites that a student has already satisfied and build on one another. However, learning objects in Galore are much richer with various attributes associated with them. These attributes constitute the metadata associated with a learning object and can be one of the following: text, simulation, visuals (images, graphs, figures), videos, interactive widgets, non-interactive widgets, numeric examples, symbolic example, coding examples, coding problems, self-assessment quiz, final quiz, etc. Based on these attributes different learning objects illuminate the concepts in different ways. For example, the concept of quantum superposition can be explained in an abstract textual manner in general terms, through numeric instantiations of it, through interactive widgets, or videos as well as visuals. Galore supports all of these alternate explanations and based on student preference instantiates one or more of these in a manner that appeals to a particular student.

Instantiation of the learning objects is supported by additional parameters such as object dependencies, outcome dependencies and learning object completion. The object dependencies are further parameterized by the level of dependency as strict, concurrent and optional. Default dependency is strict and a student must show proficiency in learning objects on which the current object depends on. Concurrent dependency indicates that the study of a learning object could be enhanced and deepened by the study of another learning object and therefore recommended for students if a certain proficiency level is attained or not. Optional dependencies can be waived.

Fig. 1. An example of a gamelet showing quantum communication exercise.

B. *Gamelets*

Gamelets are defined as small game-based interactive exercises that can be incorporated as hands-on activities to help students internalize the concepts learned through a learning object. These can be fully interactive 3D games with all the features of a traditional game and feedback that is tied into the Galore engine similar to other learning objects. The media rich elements of the game can be delivered using a variety of modes. For example, in a passive manner using videos and narrations but also in an active manner using avatars and oracles that perform various actions within a gamelet.

Each gamelet is equipped with its own game play objective (GPO) that are related to the objects of the unit or module that they are part of. Student progress in achieving proficiency in the GPOs of a gamelet is measured using a scoring system similar to serious games and their actions within the game scenario are analyzed to determine the deficiencies in student learning and corrective learning objects are then retrieved from the library as needed. The game play engine of a gamelet also supports adaptation and various modes of hints [18]–[20].

C. *Reflection Zones*

Reflection zones are primarily constructed using gamelets and interactive widgets where a student can simulate their past actions and watch a replay of their solutions to various problems. The student can also activate bots to construct alternate solutions that may have improved outcomes and compare them to their solutions to improve their performance and proficiency. Following the recreate and observe phase in a reflection zones, students have the ability to initiate queries of the oracle of the simulated scenarios allowing for a deeper internalization and comprehension of concepts. Students knowledge, queries and simulations are recorded in a student knowledge cache for future recall using a variety of modes including audio-visual notes.

D. Conceptualization Zones

The goal of the conceptualization zones is to take the knowledge acquired through gaming, simulation and reflection and abstract it in order to enable its application to different generalized but related scenarios. These generalizations may take the form of larger scale, more complexity or intricate structures pertaining to quantum protocols, states and network topologies, for example. Students are allowed to actively experiment and apply their previously learned knowledge onto new scenarios through gaming elements, interactive widgets, coding cells and visual simulations. The interactive elements often are equipped with audio-visual gaming scenarios that facilitate student queries with hints available when needed for solving new problems.

IV. The Galore Engine

The first step in building a library of learning objects for the Galore platform is instrumenting the learning objects with parameters and metadata. This is then used by the retrieval engine to automatically generate lesson plans as mentioned before. Finally, the learning objects, lesson plans and student performance are put into an adaptive loop to synthesize new and improved learning pathways for students.

A. Instrumenting Learning Objects

The instrumentation process of learning objects is driven by a domain expert. Each learning object is associated with local and global parameters and metadata. The local parameters may include items such as object-id, object-concepts, object-outcomes, object-prerequisites, etc. The local data object-type, can be one of – text, code, static visualizations, numeric example, symbolic example, videos, interactive and non-interactive widgets, quiz, auto-graded-simulation, auto-graded-quiz, code-IDE, code-IDE-wtests. Each object is also associated with a parameter called object-interactivity that is a boolean value, at present, and denotes whether a student can actively interact with the object or passively consume it (such as text, images, videos, etc.).

An important parameter associated with every object is object-alternatives that allows for one to substitute one learning object with another with a similar set of learning outcomes based on student performance and preference. For example, as mentioned before, the same concept can be explained using text, numeric example, symbolic example, simulation, widgets, etc.

Global parameters include unit-title, module-title, module-prerequisites, and module-outcomes.

B. Generating Lesson Plans

Galore generates lesson plans on demand using the following process.

Consistency Checking

The consistency of instrumentation of modules as described in the previous subsection is verified using properties such as: a) checking the consistency of global data across learning objects, b) all the module outcomes are met by the module, c) and checking the validity of object alternatives. Based on outcome dependencies Galore constructs a dependency graph. This graph is then analyzed for cycles and dangling edges. Next Galore checks to see if the module outcomes are a projection of outcomes of learning objects within the module and that the quizzes in the modules effectively evaluate the proficiency achieved in various associated outcomes.

Learning Object Retrieval

In order to generate lesson plans Galore uses user's inputted parameters and preferences and matches them to the parameters and metadata embedded within the learning objects. It then retrieves the relevant learning objects and any identified dependences. These are then weaved into a coherent linear chain using reverse topological sort based on the existing dependency graphs. The Galore engine chooses between alternate cells based on student's learning preferences and in the case of multiple learning objects satisfying the given constraints, produces a family of lesson plans that are then presented to the student to choose from.

Lesson Plan Synthesis

The linear chain of learning objects built using the retrieved objects is organized into a hierarchy, based on completion times, of nano-modules, micro-modules and modules. The organization guarantees that upon successful completion of these modules and relevant quizzes and exercises the student will meet the said objects of the modules. The modules are built using learning objects that match the highest preference indicated by the student (e.g. numerical example over symbolic examples). Similarly, certain learning objects are removed from the pre-existing modules if they are irrelevant to the learning objects chosen by the student or tangential to the indicated student learning style and preferences.

V. Generating Quantum Communication Lesson Plans: An Example

As mentioned before, learning objects are instrumented with metadata, which is used to retrieve relevant learning objects and synthesize lesson plans. There is a close correspondence between the metadata associated with the learning objects and the notebook cells. Local metadata for cells includes `cell-id`, `cell-concepts`, `cell-outcomes`, and `cell-prerequisites`. The `cell-type` have a 1-1 correspondence with the object-types and support exactly the same modes of interaction as the objects. The Boolean valued local data `cell-interactive` denotes whether a cell is interactive or not. A cell has to be interactive to be used to assess a learning outcome. For instance, cell depicted in Fig. 2 with the auto-grader is used to determine the outcome about users having a basic understanding of the conversion between polar and cartesian coordinates. The *auto-graded quizzes, numeric-examples, and code-IDE-wtests* cells are used to determine the learning outcomes of the modules containing these cells. The local data `cell-alternates` provides a list of alternate cells that are semantically equivalent to the current cell and are used to generate lesson plans that conform to student learning preferences. The global data members of a cell are the same as that of the learning objects. As an example, the Fig. (3a) depicts the data members with the first cell in the module 10 of *Galore*. The Fig. (3b) shows some of the other modules, viz. 9, 6, 5, 4, that are related to the module 10 along with the outcomes of the modules.

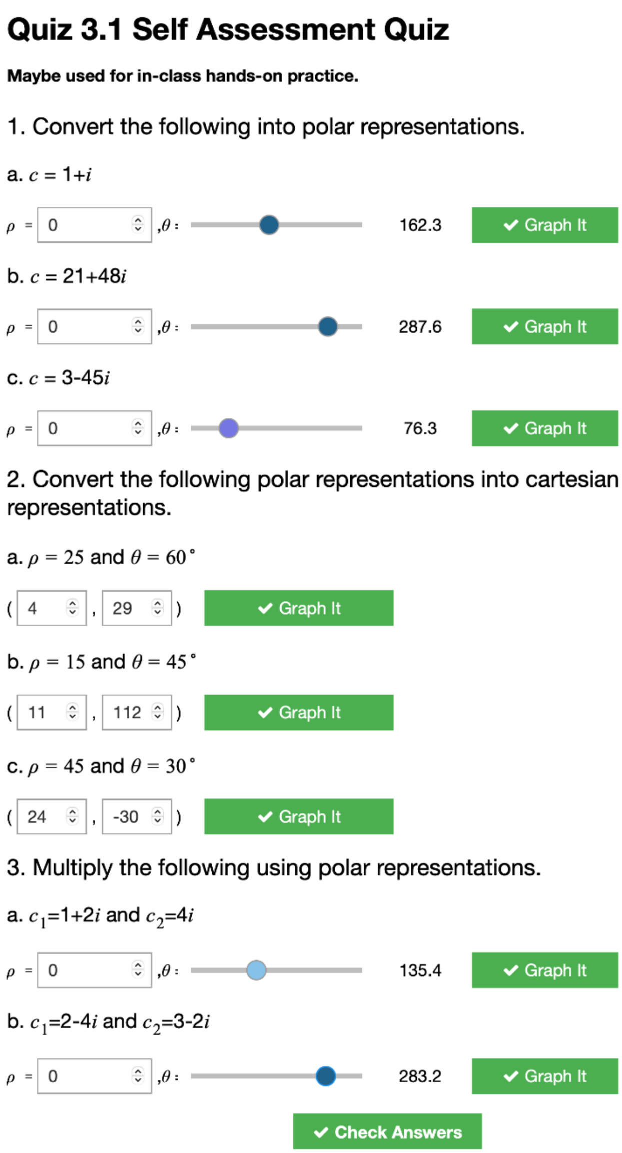


Fig. 2. A auto-graded quiz within Galore.

```
"properties": {
  "cell_ID": "m10-1",
  "cell_concepts": ["Qubit Representations", "Superposition"],
  "cell_outcomes": [
    "able to model two level quantum systems using vector and ket representation
  ],
  "cell_prereqs": [
    "understand the notions qubit, amplitude and probability of collapse",
    "be able to change between basis representations for a given qubit",
    "compute transition amplitudes from one state to the other",
    "apply the principle of quantum entanglement and Bell's inequality ..."
  ],
  "cell_type": [
    "text"
  ],
  "cell_interactive": "false",
  "cell_estimated_time": "5",
  "module_outcomes": [
    "able to model two level quantum systems using vector and ket representation
    "understand the notions qubit, amplitude and probability of collapse",
    " understand the notion of superposition",
    "be able to change between basis representations for a given qubit"
  ],
  "cell_alternatecells": [],
},
```

(a) Snapshot of a module cell metadata.

Module Number	Module Name	Module Outcomes
10	Basics of Quantum Computing/Cryptography	10.1 model two level quantum systems in vector and ket representations, 10.2 understand the notion of qubits, amplitude and probability of collapse, 10.3 understand superposition, 10.4 change basis representation
9	From Probabilistic Systems to Quantum Systems	9.1 understand probabilistic systems, 9.2 create probability vectors for probabilistic systems and apply operations, 9.3 model photons as qubits and determine their properties
6	Properties and Operations on Vectors and Matrices in Complex Vector Spaces	6.1 definition and properties of inner product, 6.2 norm of a vector, 6.3 orthogonal and orthonormal vectors and basis, 6.4 write vector using basis set, 6.5 project of vectors
5	Complex Vector Spaces: Linear Combination, Independence, Basis and Dimensions	5.1 apply concept of linear independence, 5.2 apply concepts of basis and dimensions of complex vector space

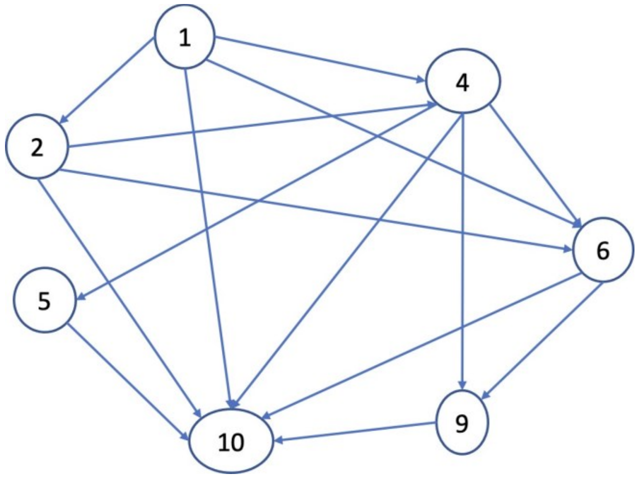
(b) Modules and corresponding outcomes.

Fig. 3. Metadata in Learning Objects

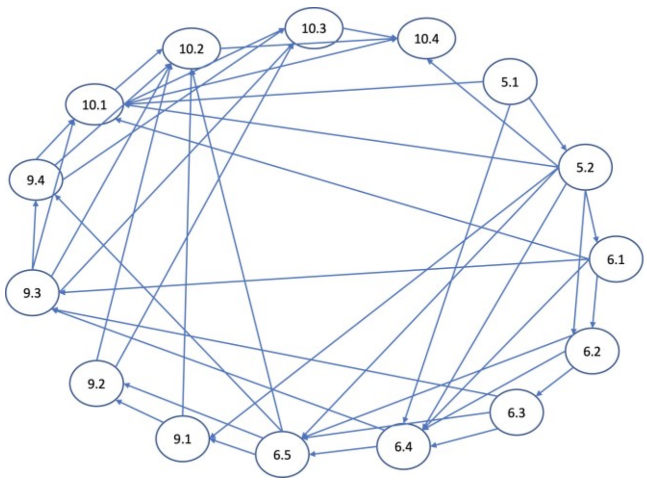
A. A Lesson Plan: Quantum Superposition

The concept of quantum superposition is central to quantum communication, computing and cryptography. It requires the understanding of complex vector spaces, linear algebra and notion of measurement. No useful computation can take place on a quantum computer or security provided in quantum communication without superposition. Because of the importance of the notion of superposition we have chosen module 10, Basics of Quantum Computing / Cryptography, as the running example.

Consider two outcomes of the module, 10.1 model two level quantum systems using vector and key representations and 10.2 understand notions of qubits, amplitudes and probability of collapse (measurement). These outcomes have dependencies in modules 1, 2, 4, and 9 (see Fig. 4a). Fig. 3b shows some of the modules and their outcomes.



(a)



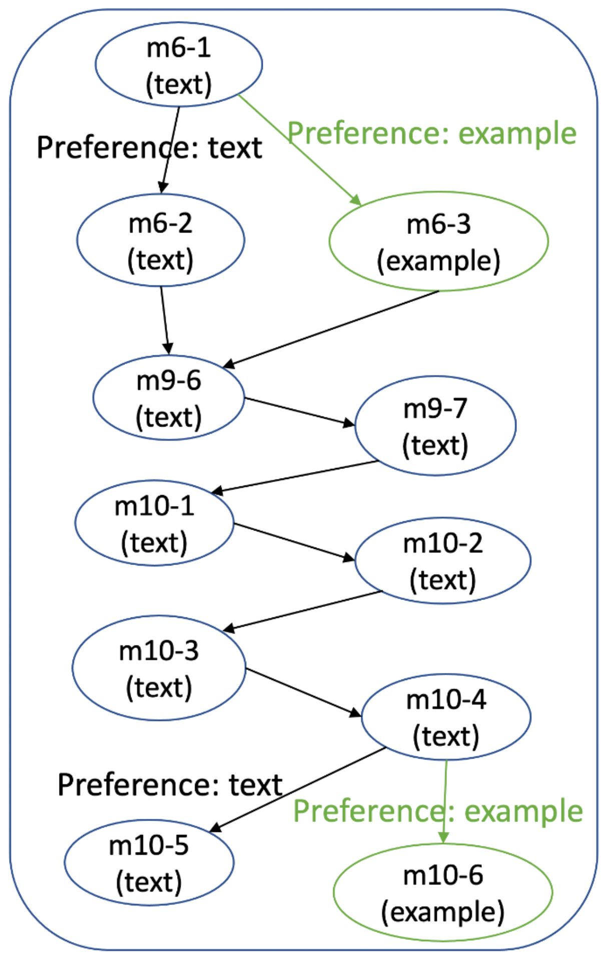
(b)

Fig. 4. (a) Module dependency graph, (b) Outcome dependency graph for outcomes 10.1-10.4. at cell level detail. (Note that all the dependencies including transitive dependencies are shown in the graph for clarity.)

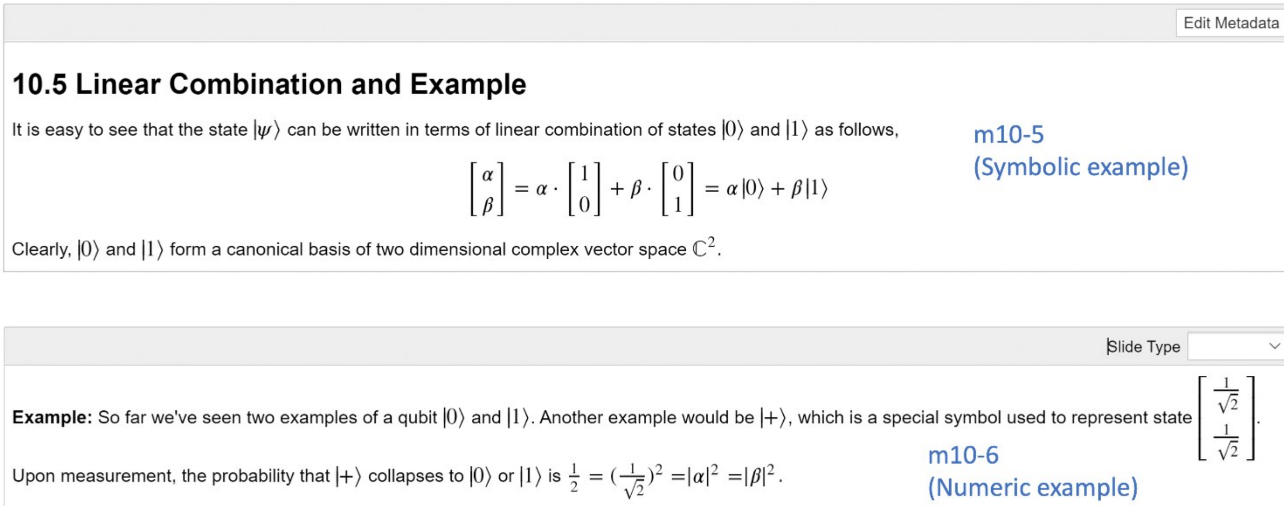
To generate a lesson involving these modules, first they are instrumented and sanity checks mentioned above were performed. Next, the module and outcome dependency graphs, depicted in Fig. 4a and Fig. 4b were generated and statically verified for the properties described above to obtain well-formed lessons and graphs.

Inputs consisting of outcome 10.2, lesson mode specifying modules as building blocks, text cell-type has the highest preference and a novice user background are used to generate a lesson that includes all of the cells of module 10 along with those of the modules 1, 2, 4, 5, 6, 9, and 10 on which module 10 depends. The reverse topological sorting of the module dependency creates a lesson with a linear chain of modules beginning with 12 cells of module 1, followed by 10 cells in module 2, followed by 18 cells of module 4, and so on until module 10. Since it is the case that either all these cells satisfy the highest preference or there exist no alternate cells that do, none of the cells in the lesson is replaced. Finally, the generated lesson is partitioned into two nanomodules, and two micromodules based on estimated times.

Next, consider generating a lesson using the same inputs as above, but changing the lesson mode to use learning object cells instead of learning object modules as building blocks of the lessons. The outcome dependency graph in Fig. 4b is used in this case to generate the lesson, which is shown in Fig. 5a. We can also generate a lesson by changing the input to specify highest preference cell-type to be a *numerical-example* instead of *text* and this will lead to the replacement cell “m10-5” by cell “m10-6” shown in Fig. 5b which are the alternate cell of “m10-5”. Note, here m10-5 is cell number 5 of module 10. If we were to provide the preference [numerical-example > example > text], then several alternate cells will be used to generate the final lesson. The resulting linear chain in the lesson is shown in the Fig. 5a with the right branch showing preference as example. These cells are organized based on their estimated time into a single nanomodule.



(a)



(b)

Fig. 5. Alternate lesson plans for learner preference "example".

Note that building a lesson for outcome 10.2 using modules leads to a lesson with 72 cells whereas we need only 28 cells if we use learning object cells to generate the lesson, which is a saving of 44 cells.

VI. Conclusions and Future Work

Classroom integration of learning objects and serious games has been a challenging problem. In this paper, we address this problem by proposing a new platform called Galore that integrates Jupyter notebook based learning objects with small game snippets, called Gamelets, that enable reflection as well as abstract conceptualization zones conforming to Kolb's four phase learning model. A lesson plan for quantum communications is presented. In the future, we plan to expand the scope of Galore to include wider cybersecurity topics for a holistic learning experience for students.

Acknowledgement

The authors would like to thank Vidya Bommanapally for help with making figures and graphs.

References

[1] "Cybersecurity library." https://www.clark.center/home.

[2] M. S. A. et. al, "Qiskit: An open-source framework for quantum computing," 2021.

[3] M. Dondlinger, "Educational video game design: A review of the literature," *Journal of Applied Education Technology*, vol. 4(1), pp. 21 – 31, 2007.

[4] E. Simpson and S. Stansberry, "A guide to integrating cots games into your classroom, in handbook of research on effective electronic gaming in education," *Games: Purpose and Potential Education*, pp. 163 – 184, 2008.

[5] R. Van Eck, "A guide to integrating cots games into your classroom, in handbook of research on effective electronic gaming in education," *Information Science*.

[6] R. Van Eck, *Gaming and cognition, theories and practice from the learning sciences, Richard Van Eck, editor*. 01 2010.

[7] B. M. Slator, O. Borchert, L. Brandt, H. Chaput, K. Erickson, G. Groesbeck, J. Halvorson, J. Hawley, G. Hokanson, D. Reetz, and B. Vender, *From Dungeons to Classrooms: The Evolution of MUDs as Learning Environments*, pp. 119–159. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[8] M. B. Nunes and M. McPherson, *Why Designers cannot be Agnostic about Pedagogy: The Influence of Constructivist Thinking in Design of e-Learning for HE*, pp. 7–30. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[9] M. K.L., A. Orr, P. Frey, D. R.P., V. V., and A. McVay, "A literature review of gaming in education," *Gaming in Education*, 2012.

[10] M. Dondlinger, "Flow," *Journal of Applied Education Technology*, vol. 4(1), pp. 21 – 31, 2013.

[11] H. Thorkild, "Game-based teaching: Practices, roles and pedagogies," *New Pedagogical approaches in Game Enhanced Learning: Curriculum Integration*, pp. 81 – 101, 2013.

[12] S. de Freitas, M. Ott, M. Popescu, and I. Stanescu, "New pedagogical approaches in game enhanced learning: Curriculum integration," 2012.

[13] E. Melis, E. Andres, J. Budenbender, A. Frischauf, G. Goduadze, P. Libbrecht, M. Pollet, and C. Ullrich, "ActiveMath: A Generic and Adaptive Web-Based Learning Environment," *International Journal of Artificial Intelligence in Education (IJAIED)*, vol. 12, pp. 385–407, 2001.

[14] H. Al-Chalabi and A. HUSSEIN, "Ontologies and personalization parameters in adaptive e-learning systems: Review," *Journal of Applied Computer Science Mathematics*, vol. 14, pp. 14–19, 01 2020.

[15] Vytautas Štuikys, Renata Burbaitė, Kristina Bespalova, and Giedrius Ziberkas, "Model-driven processes and tools to design robot-based generative learning objects for computer science education," *Science of Computer Programming*, vol. 129, pp. 48 – 71, 2016. Special issue on eLearning Software Architectures.

[16] "MERLOT." https://www.merlot.org/merlot/.

[17] V. Rossano, M. Joy, T. Roselli, and E. Sutinen, "A taxonomy for definitions and applications of los: A meta-analysis of icalt papers," *Educational Technology Society*, vol. 8, pp. 148–160, 01 2005.

[18] V. Bommanapally, M. Subramaniam, P. Chundi, and A. Parakh, "Navigation hints in serious games," in *Immersive Learning Research Network*, June 2018.

[19] D. Abeyrathna, S. Vadla, V. Bommanapally, M. Subramaniam, P. Chundi, and A. Parakh, "Analyzing and predicting player performance in a quantum cryptography serious game," in *Games and Learning Allianc*e (M. Gentile, M. Allegra, and H. So¨bke, eds.), (Cham), pp. 267– 276, Springer International Publishing, 2019.

[20] A. Parakh, M. Subramaniam, P. Chundi, and E. Ostler, "A novel approach for embedding and traversing problems in serious games," in *Proceedings of the 21st Annual Conference on Information Technology Education*, SIGITE '20, (New York, NY, USA), p. 229–235, Association for Computing Machinery, 2020.