Adaptive Learning using Finite State Machine Logic

Muffie Wiebe Waterman
Cisco Systems
San Jose, CA USA
muwiebe@cisco.com

Dennis C. Frezzo
Cisco Systems
San Jose, CA USA
dfrezzo@cisco.com

Michael X. Wang
Vitapoly, Inc.
San Francisco, CA USA
michael@vitapoly.com

ABSTRACT
We demonstrate the feasibility of Finite State Machine (FSM) logic to design adaptively scaffolded activities, presenting early work integrating adaptive learning into a learning tool in widespread use globally. We describe how integrating FSM logic with existing assessment architecture enables us to extend from direct measurement to scaffolding and feedback interventions on a spectrum of timescales from 1-second to several hours. Four prototypes are shared, demonstrating how this FSM logic affords design across differing learning contexts. Implications for design of efficiency and empowerment at scale, potential for content co-creation, and transformation of learning are discussed.

Author Keywords
adaptive learning; finite state machine; educational games; scaffolding; simulations; networking; student agency

CSS Concepts
• **Applied computing**~*Interactive learning environments*

INTRODUCTION
2020 has arrived with a thud: No flying cars, socioeconomic disparity in student performance stubbornly intact, and skills gaps widening. While competencies required for workplace success are changing rapidly [11], education infrastructure is not. Large-scale, technology-mediated learning environments offer great potential, if they can be designed to foster and assess more active forms of learning that will prepare students for future work. Researchers and practitioners are looking to new methods of instruction and new frameworks to achieve greater student participation, agency, and effectiveness in learning at scale [e.g., 6, 7, 9].

In this work-in-progress paper we present prototyped improvements to a tool used by millions of students to learn technical skills and improve their career prospects. We first describe the context of this tool's integrated assessment architecture, which supports the direct measure of student behavior and subsequent aggregation into an evidentiary chain of inference to gain a picture of student proficiency. We then present our initial progress integrating adaptive learning into the tool, where we consider *adaptive learning* as a computer-based system in which evidence of students' prior knowledge, skills, and abilities is used to determine their next activity. As Sottilare et al. [10] note, an important extension for adaptive learning at scale will include moving beyond near-term modelling of student performance, to an ability to capture and capitalize on student performance across multiple timescales. We show prototypes that enable intervention at timescales ranging from moments of task completion to hours of gamification, with the potential to reshape learning. Through ITS, games, and VR examples we introduce the underlying Finite State Machine (FSM) logic driving our adaptive learning experiences.

KEY THEORETICAL GROUNDING

Activity Theory
Activity Theory as formulated by Engeström [4] guides us toward effective design for this global student population. Three interlocking activity systems (Student, Instructor, Designer) best describe the context into which we design and test our prototypes for future learning opportunities:

Subject	*Student*	*Instructor*	*Designer*
Key Tools	*Packet Tracer (PT)* adaptive scaffolding & feedback	*PT* Sequence Editor *PT* Activity Wizard LMS	*PT* Sequence Editor *PT* Activity Wizard
Object	Master the .pks activity	Amplify .pks opportunities for practice	Develop FSM & .pks files
Goal	Learn more efficiently & with greater agency	Teach more efficiently at greater scale Empower students	Empower instructors & students for workplace of the 2020s

Table 1. Three interlocking Activity Systems informing our design of simulation-based learning environment prototypes.

Design-Based Research
We utilize rapid prototyping to develop adaptive learning tools and experiences in the domain of Information and Communication Technologies (ICT). As practitioners of design-based research [1, 2], we see alignment with Kulkarni [8] on the importance of the design perspective from which learning at scale is created; and we employ his distinction between *efficiency* (additional students an Instructor can reach) and *empowerment* (additional people available to support student learning). Here we address both perspectives,

L@S'20, August 12-14, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7951-9/20/08.
https://doi.org/10.1145/3386527.3406720

extending the concept of empowerment to include scaffolding the agentic students themselves.

LEARNING EXPERIENCE CONTEXT

Educational Program and Student Diversity

Cisco Networking Academy is an ICT educational program, freely licensed since 1997, that annually serves 2.1 million students of considerable technical and cultural diversity. Most courses are taught in person: instructors offer lectures and hands-on labs, supported by online curriculum and online formative and summative assessments. In 2019, 12,000 institutions in 180 countries offered NetAcad courses. Globally 28% of students are women.

Learning Tool

Cisco Packet Tracer is a network simulation microworld used by NetAcad students to augment the hands-on labs. Running locally on desktop, the application enables students to practice and test network behavior over a diverse array of networking devices. During 2019, over 800,000 students used *Packet Tracer*.

Previous research by Cisco established that pattern-based authoring and re-use afforded by *Packet Tracer* create an effective learning environment [5]. In addition, substantive validity was established for an educational-simulation game called *Aspire* which employs *Packet Tracer* as a backend [3].

Assessment Backbone

Packet Tracer has an extensive embedded assessment architecture, capable of measuring student actions with high fidelity and to a fine-grain level of precision. Graded activities (called .pka files) can capture all relevant gradable aspects for a specified set of networking devices or network conditions, structured as an assessment 'tree.' Typically .pka files are created using an automated Activity Wizard: the author specifies the desired scenario and the correct end-state for device configuration, and the system determines all possible observables a student will need to generate. Authors can include various learning scaffolds, including, for example, instructions, timing, and modifying or disabling student access to the assessment tree.

Evidentiary Chain of Inference to Proficiency

With *Packet Tracer's* automated grading of these direct observables, instructors can see with precision what the student did. Often this level of granularity isn't optimal for understanding student skill or mastery, and so observables are combined through three levels of increasing complexity up to Student Model Variables. From this evidentiary chain we can draw inferences from direct student behavior, defining learner *states* as proxies for student proficiency.

ADAPTIVE LEARNING – EMERGING WORK

A limitation of the *Packet Tracer* learning tool is that student feedback opportunities are sparse. Currently, if a student wants to see how they are doing on a .pka activity, the primary method available is clicking a 'Check Results' button to reveal the assessment tree. Observables already demonstrated show a green ✓; uncompleted observables have a red ✗. The student can return to their network topology knowing what steps they are missing, but not how to accomplish them. At present, there is a limited mechanism for feedback about observable-level actions, and no feedback persists across .pka activities. Observables can be combined, however, affording some inferences about proficiency.

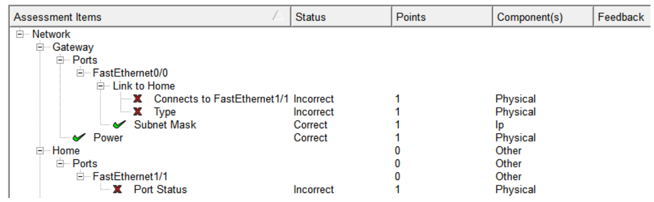

Figure 1. *Packet Tracer* assessment tree chain of inference, as seen when you click Check Results.

Efficiency and Empowerment from Adaptive Learning

Kulkarni [8] posits that designing for efficiency is often driven by a desire to bring increased access to content-specific resources at lower cost, while designing for empowerment may serve other kinds of educational goals, for example agency, creativity, or collaboration.

Our design of adaptive learning within *Packet Tracer* does have an efficiency at scale goal: we are motivated by replicating instructors' 'over the shoulder' coaching. Far from replacing classroom practitioners, our vision for this augmented support is to free up Instructor time for the creative, higher-order support that is uniquely valuable to their students. We see efficiencies, too, in decreasing the time-to-feedback for learners, based on their actions.

Efficiency is not our only goal, however. Intentionally designing for empowerment, we seek to create new computational scaffolds that can help the diverse NetAcad student population develop more agency as they learn problem solving skills in the networking domain. We also see the tool empowering teachers to provide nuanced support to the students who need it most, thereby increasing students' learning resources in the classroom. Instructors who have tested our prototypes have instantly seen applicability for student collaboration and new shared learning opportunities– skills that map well to valuable workplace competencies.

Activity Sequencer Editing Tool

Our team has built an experimental version of *Packet Tracer* using an agile software development methodology consonant with educational design-based research [9]. Specifically, we have developed a new activity editing tool, *Activity Sequencer*, comprised of an editor mode for authoring and player mode for student learning experience.

The *Activity Sequencer* applies Finite State Machine (FSM) logic to authoring a networking activity. Authors can define **states** of the experience and **triggers** for **transitions** between states. States are an evidence-based inference of the learner's state of proficiency, derived from the aggregation of observables in the assessment tree. Triggers for transitions are based on a combination of the learner's near-term performance plus prior behavior and success within the sequence, in real-time response to current observables. We are exploiting the FSM logic in a variety of learning contexts, including a simple Intelligent Tutoring System (ITS), single-player and multiplayer educational games, Computer Adaptive Testing (CAT), and as a backend to Virtual Reality (VR) environments and games.

The flexibility of the FSM design allows us to apply observables in multiple ways, as part of FSM states and transitions. The same observable can be applied differently, meaning that the FSMs can react differently to the same observable. For example, the direct observable of a specific IP address could be used to trigger a hint in an ITS or to declare that a contract is done in a game. In addition, by incorporating *Activity Sequencer* FSM logic into *Packet Tracer's* existing Activity Wizard, we can create scaffolded learning experiences of differing timescales, from within a single .pka activity to across multiple .pka files. These adaptive activity sequences (called .pks files) allow for new forms of responsive scaffolding, from hints, feedback, and activity choices. Lastly, FSM logic supports versatile design of story settings, creating the potential to engage more student personas and inspire more student agency. Below we discuss some of our prototypes in ITS, games, and VR.

Intelligent Tutoring System (ITS) example – Default Gateway

We have created an adaptive learning .pks file for a common student difficulty: setting up a network so that a user device can successfully connect to the internet (configuring the default gateway). Students are given an incomplete network topology and must determine what is needed, make changes to the network, and use a ping command to verify they are able to connect to a mock web server inside the simulation.

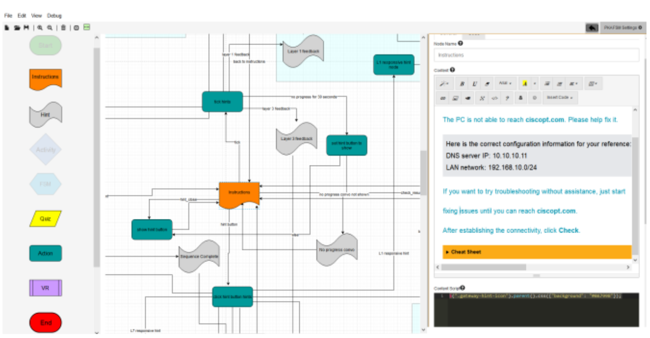

Figure 2. ***Activity Sequencer*** **editor view of Default Gateway.**

The classic lament from students stuck troubleshooting this is *"My ping failed!"* Instructors can work with the student, assessing their current work and asking questions to scaffold them toward a solution. Transferring 20-30 instances of this low-level guidance to an ITS frees up the instructor to go deeper with students who need more. A student's changes to the network are tracked as observables–from simple observables like the wrong default gateway address on a PC, to more complex observables like a ping test that fails. These observables can be defined as FSM triggers for increasingly specific hints, starting with what principle applies and ending with a bottom-out hint for the required step, affording students more tailored feedback.

Game example 1 – Aspire revisited

Inspired by some game designs which use FSM logic to create in-game adaptivity, as a proof of concept we have used *Activity Sequencer* to recreate part of the educational-simulation game, *Aspire*. Players are positioned as networking professionals taking on small contracts, and they choose which contracts to complete and how to solve them. Unique .pka files for each contract are connected by FSM state transitions, triggered by combinations of observables.

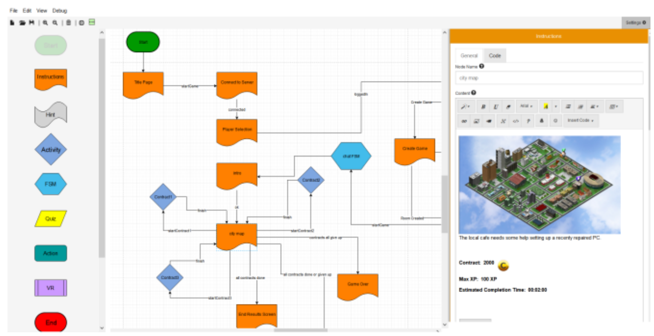

Figure 3. ***Activity Sequencer*** **editor view of** ***Aspire.***

Game example 2 – Rapid Fire

We have developed a real-time multiplayer game in which players compete to solve small networking tasks quickly. Tasks in *Rapid Fire* require configuring, troubleshooting, or design skills; a round ends when either player beats the clock, or time runs out. Fast-paced and playful, the game is a highly engaging learning experience. We have designed three .pka files as game levels connected by FSM logic in a .pks file, so that players can level up as they demonstrate skill mastery.

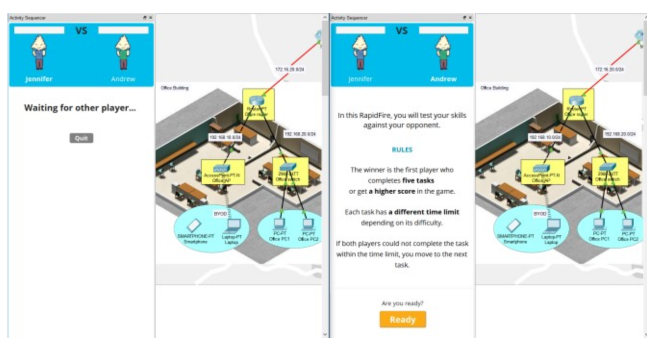

Figure 4. ***Activity Sequencer*** **player views of** ***Rapid Fire.***

Virtual Reality (VR) examples

We are developing immersive VR learning experiences in which students can explore both real-world physical networking tasks (racking, cabling, and configuring equipment) and 'magical' abstractions of networking concepts (acting as a routing algorithm forwarding packets). We have integrated *Packet Tracer* as a backend with a frontend VR UI. Using the FSM-enabled *Activity Sequencer* we can provide scaffolding and assessment across the full depth of the networks modeled in the .pks file. We are investigating VR for engagement, exploration, practice, and assessment in networking education.

CONCLUSION

The key insight driving our prototyping is that we can treat moments of student proficiency as states to be acted on: using a Finite State Machine (FSM) logic, we have designed activity levels as proxies of learner proficiency states.

This FSM editor has proven highly generative, enabling us to design for a wide range of learning experiences:

- **Four distinct adaptive learning contexts**: Intelligent Tutors, Games, Computer Adaptive Testing, and VR
- **Different timescales for intervention**: at the 1-second interval within a single activity or hours across activities
- **Single and Multiplayer instances**

In our work, the FSM is coupled with a high-fidelity, constructivist learning environment for ICT. By enabling a form of level design, the FSM editor and player empowers student agency in networking problem solving: the system can respond to where the student is at a point in time, informed by evidence collected across the activity sequence.

Packet Tracer's current Activity Wizard enables authors to design networking activities across myriad devices and configurations. Designing for empowerment at scale, we envision *Activity Sequencer* could be productized as the next level of the Activity Wizard. This would put *authoring* capability for adaptive learning directly into the hands of instructors and potentially even students, opening vast opportunities for co-creation and sharing of content.

FUTURE DIRECTIONS

Early user testing of the adaptive learning ITS and game prototypes with sixty NetAcad instructors confirms there is strong interest in the field, particularly for multiplayer gaming. Instructors reported applicability for empowering their students' collaboration in skills development, for exam practice and for increased social connection. Consonant with Kross and Guo's [7] vision of fruitful future learning@ scale research, we seek to better understand how diverse, global NetAcad students and instructors may employ these kinds of adaptive learning experiences. We see much potential of the prototypes scaling for *efficiency*, e.g., in rural environments, and also for *empowerment*, e.g., how storytelling contexts may support agency and other workplace competencies for differing student personas. In the near term, we anticipate user testing with students this year. In the longer term, we can envision complementing the current Subject Matter Expertise scaffolding with Machine Learning in the cloud. We envision a time when hundreds of thousands of students using *Packet Tracer* can go beyond hitting Check Results, to obtain more responsive and empowering feedback. The possibilities for transforming learning are substantial.

ACKNOWLEDGMENTS

We are especially grateful to the NetAcad Instructors whose invaluable feedback has driven our design iterations.

REFERENCES

[1] Ann L. Brown. 1992. Design experiments: theoretical and methodological challenges in creating complex interventions in classroom settings. *The Journal of the Learning Sciences*, *2*(2), 141-178.

[2] Allan M. Collins. 1992. Toward a design science of education. In E. Scanlon & T. O'Shea (Eds.), *New Directions in Educational Technology*, 15-22. Berlin: Springer Verlag.

[3] Kristen E. DiCerbo, Dennis C. Frezzo, and Tony Deng. 2011. Substantive validity of a simulation-based game. *Research and Practice in Technology Enhanced Learning*, *6*(3), 161-185.

[4] Yrjö Engeström. 1987, 2015. *Learning by Expanding: An Activity-Theoretical Approach to Developmental Research.* Cambridge Univ. Press.

[5] Dennis C. Frezzo, John T. Behrens, and Roy J. Mislevy. 2010. Design patterns for learning and assessment: facilitating the introduction of a complex simulation-based learning environment into a community of instructors. *Journal of Science Education and Technology, 19*, 105-114. doi.org/10.1007/s10956-009-9192-0

[6] David A. Joyner, Ashok K. Goel, and Charles Isbell. 2016. The unexpected pedagogical benefits of making higher education accessible. In *Proceedings of the Conference on Learning @ Scale (L@S'16),* 1-4.

[7] Sean Kross and Philip J. Guo. 2018. Students, systems, and interactions: synthesizing the first four years of Learning@Scale and charting the future. *Proceedings of the Conference on Learning @ Scale (L@S'18),* 1-10.

[8] Chinmay Kulkarni. 2019. Two views of scale: design principles for scaling reach and empowerment. *Proceedings of the Conference on Learning @ Scale (L@S'19),* 1-11.

[9] Denise Nacu, Pooja Upadhyay, Evan Skorep, Tre Everette, Evelyn Flores, Mighel Jackson, and Nichole Pinkard. 2019. Implementing learning analytics to foster a STEM learning ecosystem at the city-level: emerging research and design challenges. *Proceedings of the Conference on Learning @ Scale (L@S'19),* 1-4.

[10] Robert A. Sottilare, Rodney A. Long, and Benjamin S. Goldberg. 2017. Enhancing the eXperience Application Program Interface (xAPI) to improve domain competency modeling for adaptive instruction. *Proceedings of the Conference on Learning @ Scale (L@S'17), 1-4.*

[11] World Economic Forum. 2018. *The Future of Jobs Report 2018.*