Psychometric and Evidentiary Approaches to Simulation Assessment in Packet Tracer Software

Dennis C. Frezzo John T. Behrens Robert J. Mislevy Patti West Kristen E. DiCerbo
Cisco Cisco &Univ. of Notre Dame[1] University of Maryland Cisco Cisco Learning Inst.
dfrezzo@cisco.com jbehrens@cisco.com rmislevy@umd.edu pawest@cisco.com
kristen.dicerbo@ciscolearning.org

Abstract

Packet Tracer is a comprehensive instructional software application for teaching skills and concepts associated with computer networking. In addition to a wide range of simulation, visualization, and micro-world authoring features to support student-centered Exploration, Explanation and Experimentation, Packet Tracer includes features for assessment task authoring with automated scoring and reporting. This paper provides a theoretical explanation for these features following the approach of Evidence Centered Design (ECD) and discusses implications for future extension. It is argued that the four process model for assessment delivery provides a number of valuable features that allow for conceptual and computational extension into improved simulation-based assessment, as well as gaming and tutoring.

1. Introduction

The Cisco Networking Academy program is a collaborative effort between Cisco, the Cisco Learning Institute, and educational institutions around the globe. The goal of the program is to promote the learning of knowledge and skills associated with computing and computer networking. To support this effort Cisco provides a number of tools to instructors including on-line curriculum, on-line assessment [1] and Packet Tracer software. Packet Tracer (PT) is a comprehensive simulation, visualization, collaboration, and micro-world authoring tool for teaching networking concepts.

1.1 Packet Tracer Conceptual Levels

Packet Tracer can be thought of as providing instructional and assessment services at a number of levels [2]. At the most obvious level, PT provides a comprehensive Cisco IOS and PC network simulation. The behavior of a range of protocols is spans from RIP to multi-area OSPF. This allows a wide range of practice and exploration. In addition, a number of packet visualization interfaces are provided to help learners visualize difficult to understand concepts. At the second layer of the hierarchy, PT provides a graphical interface that allows the designing and building of networks by simple drag-and-drop functions combined with the underlying simulation layer. At a third layer, authoring features are available to save and lock networks, and add stories or task requirements for games or assessments. This allows instructors to create their own network mini-worlds (in their local language) and control the features described above.

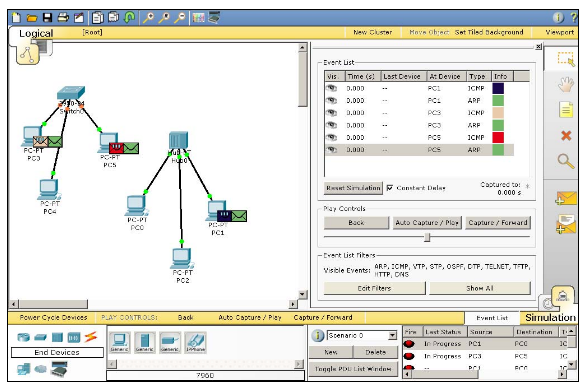


Figure 1. Screen Shot of Packet Tracer Software

A unique micro-world assessment authoring interface and a corresponding variable manager complete the educational layers of the current version of PT. A screen shot of PT is provided in Figure 1. In this figure, the reader can see the software providing detailed information regarding network traffic (right) while corresponding packets are shown traversing the network (left). Networks are authored by dragging devices from the palettes in the lower left and interacting with additional interfaces that can appear in the network diagram when device icons are double-clicked.

The framework of the PT instructional affordances provides students and instructors with:

- Exploratory, self-directed learning opportunities
- Demonstrative activities that illustrate important concepts
- Structured practice activities to guide learning
- Assessment opportunities to gauge progress and provide feedback

While all these affordances have value in themselves, the assessment infrastructure is of particular interest because its computation requires a detailed analysis of the structure of the environment as well as the student-task interaction, and an understanding of the pedagogical structure of the network micro-world.

Because Evidence Centered Design (ECD) [3-5] provides a broad articulation of structural and process aspects of assessment and related educational activities, it has become our core conceptual model. In the following sections we (1) introduce the central concepts of ECD, (2) explain how the map into current PT features and structures, and (3) describe how they position us for possible future extension and innovation.

2. Evidence Centered Design and the Four Process Delivery Model

Evidence Centered Design provides a series of descriptive models that enumerate components of a broad range of assessment undertakings and is especially valuable for describing the structure of complex environments and tasks such as those provided in a complex simulator [2].

In our setting, assessment is considered the process of characterizing learner knowledge, skills, and abilities. Subsequent uses of these data for program or individual evaluation are separate functions from the core goal we describe here.

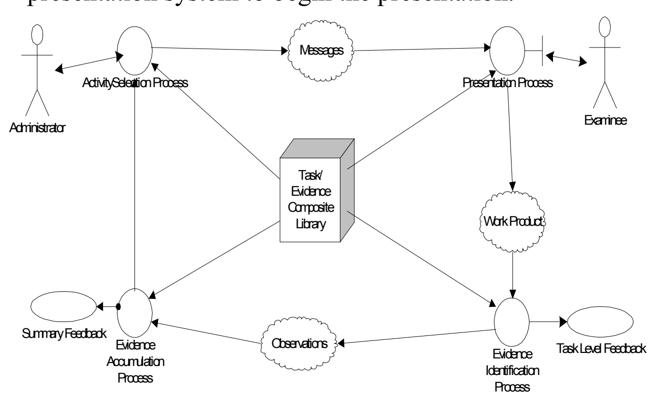
The process of developing an assessment to inform us about student progress and proficiency within a specific domain begins with up-front analysis and design. The first step in this process includes an analysis of the structure of the domain, specification of the purpose and intended use of an assessment and the claims regarding the knowledge skills or abilities one wants to make about the individuals. After these essential preliminaries are established, the assessment design process can be thought of in terms of a student model, evidence model, and task models. Student models represent the areas of student proficiency that the assessment is intended measure. Evidence models describe what student behaviors or artifacts can be used to indicate these proficiencies and how they are to be interpreted. Task models define methods to elicit the specified student behaviors or artifacts

For the purposes of this paper, we will focus on a different projection on the process in terms of the four-process delivery model which uses the three models mentioned above.

The ECD Four Process Model is depicted in Figure 2. The model consists of four primary processes: Task Selection, Presentation, Evidence Identification, and Evidence Accumulation. Each of these processes relies on data depicted in the center of the diagram.

2.1 Task Selection

The task selection process calls to our attention the fact that assessment experiences are composed of tasks and the choice of a task must be made. In common multiple-choice assessment systems, the task selection is typically "present the next question", however in adaptive testing the task selection may be made based on intelligent or adaptive systems that optimize the information function or other variable in the system [6]. The output of this system is the message to a presentation system to begin the presentation.



2.2. Presentation

The presentation system is responsible for interacting with the learner and allowing the construction of a work product. A work product is "a unique human production-a mark on an answer sheet, written directions from the hotel to the bank, a sequence of utterances…"[3]. The idea here is that work products are designed beforehand to capture the relevant outcomes of the task. Instructors often use router logs and final configuration files as the work products for in-class tests or observations. Because PT has access to all state information in the micro-world network it has allowed the user to create, a series of network-state variables comprises the work product in

PT. Conceptually, the final network and its functionality are the work products.

2.3. Evidence Identification

The third process is called evidence identification. The purpose of this process is to apply rules to the work product to create observable variables. In the common multiple choice format the rules are built into the format so that typically a single rule is applied that "if examinee answer = correct answer" evidence of proficiency is determined. Of course, more complex scoring could occur here including having different observables for different answer choices.

Unique challenges arise in the specification of evaluation rules in an open simulation environment. First, while complex rules could always be authored by direct computer programming, we desired a system that could be used by instructors and students with a (relatively) easy to use interface. Second, as the simulation environment grows, the number of possible features of interest and their combinations grows exponentially. Third, as the complexity of the system grows, the complex of observable variables and their linkages to final reporting variables grows in complexity as well.

2.4 Evidence Accumulation

The fourth process is evidence accumulation. This is the process of combining observable variables according to rules and weights to provide a final descriptor of the individual's state of knowledge, skill or ability. Computationally, these variables are called Student Model Variables or SMVs. In computationally refined systems, this process is probabilistic, recognizing that each single data-point is an imperfect measure, and summary variables bring forth those vagaries. In practice, however, most educational systems (including classroom instructors) combine evidence deterministically, and create mental probabilistic adjustments to the final outcomes post-hoc. That is to say, the instructor often adds up all the points on the exam (perhaps weighting some more than others) and concludes "yes the student got 90% correct, and I think therefore highly proficient, but I know that is not a perfect measure". In many traditional assessment systems each task (sometimes called items) generates one observable and it loads on one SMV. Such an approach is extremely limiting in a simulation environment because some observables may provide important information (perhaps differentially weighted) to different SMVs. For example, a data bit identifying a properly configured router (an observable variable) may provide information both for SMVs regarding configuration skills, as well as trouble shooting skills if the outcome occurred in the context of a troubleshooting task.

3.0 Application of ECD in Packet Tracer

3.1. Task Selection

While Packet Tracer's affordances for simulation-based assessment are extremely strong, it has been developed as a general simulation, visualization and authoring tool for many uses. Accordingly Packet Tracer is designed to fit into a number of larger systems. For example, Packet Tracer activities (with or without assessment feedback) are built into the current curriculum experience in the Cisco Networking Academy Program. Given this flexibility, Task Selection is assumed to occur outside of the Packet Tracer system itself. In the ECD model, PT assumes a task is given (by loading a network and task file) and that PT will act over the remaining three processes.

3.2. Presentation

The rich interface and drag-and-drop interaction of PT is a hallmark of the software. As discussed above, it provides a deep (though imperfect) simulation of a broad range of networking devices and networking protocols including rich features set around the Cisco IOS. Instructions for tasks can be presented through HTML formatted text boxes which can be pre-authored and locked by any user. In this way Packet Tracer is not simply a simulation tool, but actually a micro-world authoring and interaction tool with instructional and assessment affordances. The availability of this tool to Cisco Networking Academy students around the world provides increase practice and access to network configurations unobtainable for many students and instructors.

One important differentiator between PT and many other instructional environments is that it has a variable manager feature that allows the template creation of networks and the generation of random version of the network based on ranges of values. Values can be generated by random selection from lists or numeric ranges in both the textboxes that describe the task or activity, as well as in values of much of the network data (e.g. IP addresses). This allows for the development of large number of practice examples or isomorphic tasks generated at run time.

3.3. Evidence Identification/Task Scoring

In light of the difficulties noted in section 2.3, the approach taken in PT to help with the authoring of scoring rules needed to be easy to use, comprehensive, and consistent with the flexibility of the ECD model. To accomplish this, PT allows the construction of an "Answer Network" by providing the author with a comprehensive list of network states as depicted in the tree on the left side of Figure 3. The original state of the "Answer Network" provides a list of potential work product features of interest to the assessment designer. These are low-level features including whether a particular interface is up, whether a cable is plugged in, and the value of an IP address on each interface. By considering the purpose of the assessment and the relationship between work product features and signs of proficiency, the designer checks the boxes of all the features about which they would like to obtain data.

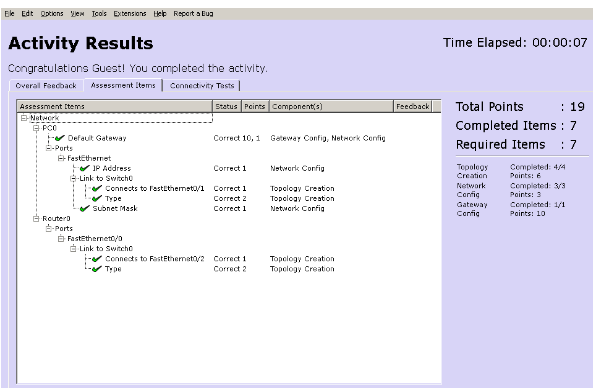


Figure 3. Packet Tracer Results screen showing features of Answer Network (left), observable values, component loading values, and components.

After indicating the features about which he would like to create observable variables, the assessment designer edits the answer network values to create implied scoring rules. At time of task completion, the target features of the learner network are compared pair-wise with the corresponding features of the answer network. The comparisons used to generate observable variables may look for exact matches (answer network cable plugged in, examinee network cable plugged in) or may allow for evaluation of range of values (answer IP address between xx.xx.0 and xx.xx.100, examine network IP address xx.xx.50.). This pattern matching is made flexible by both the use of the variable manager and the use of regular expressions in the evaluation clauses. Network functionality tests can also be authored in addition to answer network end-state tests. While log files are not currently being evaluated by the PT scoring engine, they are accumulated and available for further review by students and instructors, The files can also be exported to automated systems for analysis and evaluation.

Another feature of Evidence Identification in the ECD framework is that it is the minimum requirement for providing feedback. After some aspects of the work products have been characterized into observable variables, these variables can be used as the triggers for feedback to the learners. Packet Tracer supports this by allowing the authoring and reporting of observable level feedback. That is to say, different strings can be presented to the learner depending on whether a specific work product feature is present or not.

In the results window presented to the student (as seen in Figure 3), the value of the Observable Variables are communicated with the labels of "correct" and "incorrect". This is conducive to communicating to students but is not the only possible understanding. From and ECD perspective, the key point is that the value of the feature in the work product matches or does not match a feature in the evaluation key – the Answer Network.

3.4. Evidence Accumulation

While Packet Tracer was not designed primarily as an assessment and measurement tool, it has important features that support the facilitation of linkages from observables to variables called "components" which serve as SMVs in the ECD framework. For each observable variable, PT allows the specification of multiple components to be associated with the observable variable and allows the specification of differential weights. In the ECD model this is described as a multiple-observable/multiple-student model variable architecture.

An important concept is that the different observable variables can provide information in multiple dimensions. Accordingly it is important to conceptualize the observable variables not simply as identifiers of correctness, but rather as a piece of information about a feature that provides information for one or more SMVs. In many traditional assessment systems each task generates one observable and it loads on one student model variable.

The primary limitation of the Packet Tracer software in this area is that loadings between the observables and the components (SMVs) are limited to integers combined with summation. One possibility is that future versions could allow probabilistic weightings that could be obtained using external analysis such as the application of Item Response Theory or Bayesian Inference methods [7].

4. Implications and Future Possibilities

In the previous sections we have illustrated how the PT assessment infrastructure has leveraged the flexibility of the ECD Four Process Model to provide a number of detailed object layers to promote assessment infrastructure. Indeed, we believe PT is unique in enabling a wide range of authors (professional designers, instructors, students) to create their own simulation-based micro-worlds and to construct assessment models with automated scoring following a multiple-observable/multiple-SMV framework.

Nevertheless, a number of limitations remain. First, while the assessment authoring interfaces allow the identification and scoring of a wide range of low-level observables (e.g. is the link light on), there is currently no facility to combine these low-level or primary observables, into more integrated or compound observables. Assessment designers often want to work at a more aggregated level such as "was the router configured properly", rather than "was the IP address in such a range"? To address this, discussions are currently underway concerning the possibility including a small macro language that would allow the specification of compound observables as the result of logical and mathematical operators on the values or primary observables. This would allow observable variables to be created using logic such as "if the cable is plugged in but the power is off then….". Such disjunctive rules are not directly supported in the current interface.

A second area of limitation is in the external task-selection assumption. As note above, PT can be used to open pre-defined tasks or can be launched by the activation of pre-defined task that will call the executable program with which it is associated. More sophisticated and interactive PT activities could be created if there were mechanisms for real-time task selection or task modification in accordance with different optimization targets. For example, in a gaming system task selection is optimized for maximum motivation. After a tasks is complete, the SMVs are updated and a new task is chosen that is optimized to keep interest high and game playing continual. In an adaptive testing situation, task selection is often built to optimize a test information function. After the appropriate SMVs are updated, the subsequent tasks are selected to maximize the information available about the learner.

To address this set of possibilities, research is currently underway regarding the possible use of external programs that could control PT task selection via a built-in API, similar to that which currently allows PT instances to communicate with each other for multi-player gaming. By extending this technology, we may be able to extend uses of PT to include intelligently interactive gaming, adaptive testing, or intelligent tutoring.

It has been argued elsewhere [8] that the structural, functional, and semiotic symmetries between assessment and gaming in a simulation environment are numerous and that these symmetries offer computational synergies between the two sets of goals when systems are designed with ECD capabilities. For example, both assessment and gaming systems provide tasks, score the work and provide feedback to the learner/gamer. Rich presentation (simulation) environments such as Packet Tracer are also well suited for use in gaming configurations, and a number of games are currently available. Recent addition of Packet Tracer Messaging Protocol allows multiple instantiations of Packet Tracer to communicate with each other. This enables the possibility of on-line gaming through competitive or collaborative task development.

In this paper we have attempted to show how ECD provides a comprehensive and flexible framework understanding, developing and extending evidence-based instructional architectures in complex simulation environments. This has been accomplished here through illustrations using Packet Tracer software. While at present, the breadth of all possible applications are not yet implemented, we hope to have illustrated how the flexible data and process architecture provided by ECD supports not only extensive assessment expansion, but expansion in other applications that require detailed knowledge of learner knowledge, skills, and abilities.

5. Acknowledgments

(1) John Behrens is Assistant Adjunct Research Professor in the Department of Psychology at the University of Notre Dame (Notre Dame, IN, USA), and would like to thank the department for its support.

6. References

[1] J.T. Behrens, T.A. Collison, and S.F. DeMark, "The Seven Cs of comprehensive assessment: Lessons learned from 40 million classroom exams in the Cisco Networking Academy Program," *Online Assessment and Measurement: Case Studies in Higher Education, K-12 and Corporate.*, Hershey, PA.: Information Science Publishing, 2005, pp. 229-245.

[2] Frezzo, D. C., Behrens, J. T., and Mislevy, R. J., "Design patterns for learning and assessment:

Facilitating the introduction of a complex simulation-based learning environment into a community of instructors.," *The Journal of Science Education and Technology*, in press. .

[3] R.J. Mislevy, L.S. Steinberg, and R.G. Almond, "Design and analysis in task-based language assessment," *Language Testing*, vol. 19, Oct. 2002, pp. 477-496.

[4] R.J. Mislevy, L.S. Steinberg, and R.G. Almond, "On the Structure of Educational Assessments," *Measurement: Interdisciplinary Research and Perspectives*, vol. 1, 2003, pp. 3-67.

[5] J.T. Behrens, R.J. Mislevy, M. Bauer, D.M. Williamson, and R. Levy, "Introduction to Evidence Centered Design and Lessons Learned From Its Application in a Global E-Learning Program.," *International Journal of Testing*, vol. 4, 2004, pp. 295-301.

[6] H. Wainer, N.J. Dorans, R. Flaugher, B.F. Green, R.J. Mislevy, L. Steinberg, and D. Thissen, *Computerized Adaptive Testing: A Primer*, Lawrence Erlbaum, 2000.

[7] R.J. Mislevy, "Evidence and Inference in Educational Assessment," *Psychometrika*, vol. 59, 1994, pp. 439-483.

[8] Behrens, John T., Frezzo, D. C., Mislevy, R. J., Kroopnick, M., and Wise, D., "Structural, Functional, and Semiotic Symmetries in Simulation-Based Games and Assessments," *Assessment of Problem Solving Using Simulations*, New York: Earlbaum, 2007, pp. 59-80.