

Design Patterns for Learning and Assessment: Facilitating the Introduction of a Complex  
Simulation-Based Learning Environment into a Community of Instructors

Dennis C. Frezzo<sup>1</sup>, John T. Behrens<sup>2</sup>, and Robert J. Mislevy<sup>3</sup>

Mailing address for correspondence:

Robert Mislevy

EDMS, Benjamin 1230-C

University of Maryland

College Park, MD 20742

301-405-1933

rmislevy@umd.edu

---

<sup>1</sup> Networking Academy Learning Systems Development, Cisco Systems, Inc., Mailstop SJC05/3/4, 325 East Tasman Drive, San Jose, CA, 95134.

<sup>2</sup> Networking Academy Learning Systems Development, Cisco Systems, Inc., 15300 Fox Run Trail, Mishawaka, IN, 46545.

<sup>3</sup> Measurement, Statistics and Evaluation, University of Maryland, College Park, MD, USA 20742.

Design Patterns for Learning and Assessment: Facilitating the Introduction of a Complex  
Simulation-Based Learning Environment into a Community of Instructors

Abstract

Simulation environments make it possible for science and engineering students to learn to interact with complex systems. Putting these capabilities to effective use for learning and assessing learning requires more than a simulation environment alone. It requires a conceptual framework for the knowledge, skills, and ways of thinking that are meant to be developed, in order to design activities that target these capabilities. The challenges of using simulation environments effectively are especially daunting in dispersed social systems. This article describes how these challenges were addressed in the context of the Cisco Networking Academy program (CNAP) with a simulation tool for computer networks called Packet Tracer. The focus is on a conceptual support framework for instructors in over 8000 institutions around the world for using Packet Tracer in instruction and assessment, by learning to create problem-solving scenarios that are at once tuned to the local needs of their students and consistent with the epistemic frame of “thinking like a network engineer.” We describe a layered framework of tools and interfaces above the simulation per se that supports the use of Packet Tracer in the distributed community of instructors.

Key words: Assessment, design patterns, Packet Tracer, simulation.

## **INTRODUCTION**

The Cisco Networking Academy program collaborates with over 8,000 schools in 165 countries, providing free on-line curricular and assessment materials to help teach students and instructors basic and advanced computer networking skills. To accomplish our goals we provide a toolbox of aides to the instructors and students including on-line interactive curricula, on-line assessments for chapter, mid-term and course final level feedback with an integrated on-line grade book for instructors and students, a human network of cooperating educational institutions that train instructors in partnering institutions, discounted equipment to provide hands-on instruction and assessment, global support desk and other features.

While widely adopted and receiving high customer satisfaction ratings, we recognize that these tools are primarily electronic analogs of pre-digital educational tools. The on-line curriculum is much like an animated book with fixed chapter and section structure. The on-line assessments are largely multiple-choice with immediate automated scoring, although there are some sophisticated network simulation tasks as well. In addition to communication of information and feedback, we want to provide the learners and instructors with tools that enable these agents to perform their own explanation, exploration and experimentation. We want to enable them to go beyond what we can do as curriculum and assessment designers to provide their own scaffolding and experience.

To accomplish this we have developed computer software that provides complex simulation and visualization of core features of computer networking along with

authoring features for students and instructors. This tool, called Packet Tracer, covers a broad range of representational affordances over physical and logical network topologies, myriad networking protocols and multiple layers of visualization and interactivity.

In this paper we describe our work with Packet Tracer as a user extensible and configurable micro-world that promotes instructional enablement and local customization by building off of cognitive sensitivity to the domain, and promoting design patterns in both the creation of scenarios and the development of authoring tools that promote patterned based thinking. This is accomplished in four sections. First, we provide a brief background into the instructional context driving the creation of the tool and key features of the software. Second, we discuss core design features of the software environment and its application in the field. Third, we discuss some of the cognitive underpinnings of this approach and their instantiation in design patterns to structure the dialog, communication, and instantiation of strategies. The fourth section concludes with a brief summary.

## **INSTRUCTIONAL CONTEXT**

The work discussed in this paper occurs in the context of the Cisco Networking Academy program. This program is a global endeavor to support the acquisition of Information and Communication Technology (ICT) skills associated with computer networking. Computer networks consist of devices that provide communication paths between electronic machines to create local networks inside buildings as well as the interconnection of networks in such structures as the Internet. The program is a cooperative undertaking between Cisco Systems, Inc., the Cisco Learning Institute, and approximately 8,000 public educational institutions across more than 160 countries. At present, 75% of participating academies are located outside of the United States. Cisco

Systems, Inc. provides free on-line curricula and on-line assessment to participating institutions who are required to send instructors to instructor training, monitor student outcomes, and include hands-on activities (including the use of real networking equipment) in the classroom. Instructor training and community support is provided by partnering educational institutions themselves. After completing four courses in the program, students are typically prepared to take the Cisco Certified Networking Associate ® (CCNA) exam which provides credentialing for work in business and industry.

The assessment program in the Networking Academies has provided over 60 million on-line exams across 11 languages in the last 9 years. The structure and distinguishing features of the assessment program have been described by Behrens, Collison & DeMark (2005) with additional information provided by Behrens, Mislevy, Bauer, Williamson & Levy (2004).

In most situations, students participating in the Networking Academy program read the on-line curriculum outside of class time with class used to focus on hand-on skills, elaboration of material and the use of proctored on-line assessments. Given the fact that the program is a collection of global partnerships, local conditions and adaptations vary according to a large number of factors including instructor background, student background, local customs and laws, and local types of networking businesses and design practices.

Murnane, Sharkey & Levy (2002) provide a discussion of the origin of the program, while Levy & Murnane (2004) describe issues related to technological application of the curriculum and assessment. Porter & Kramer (2002) discuss the

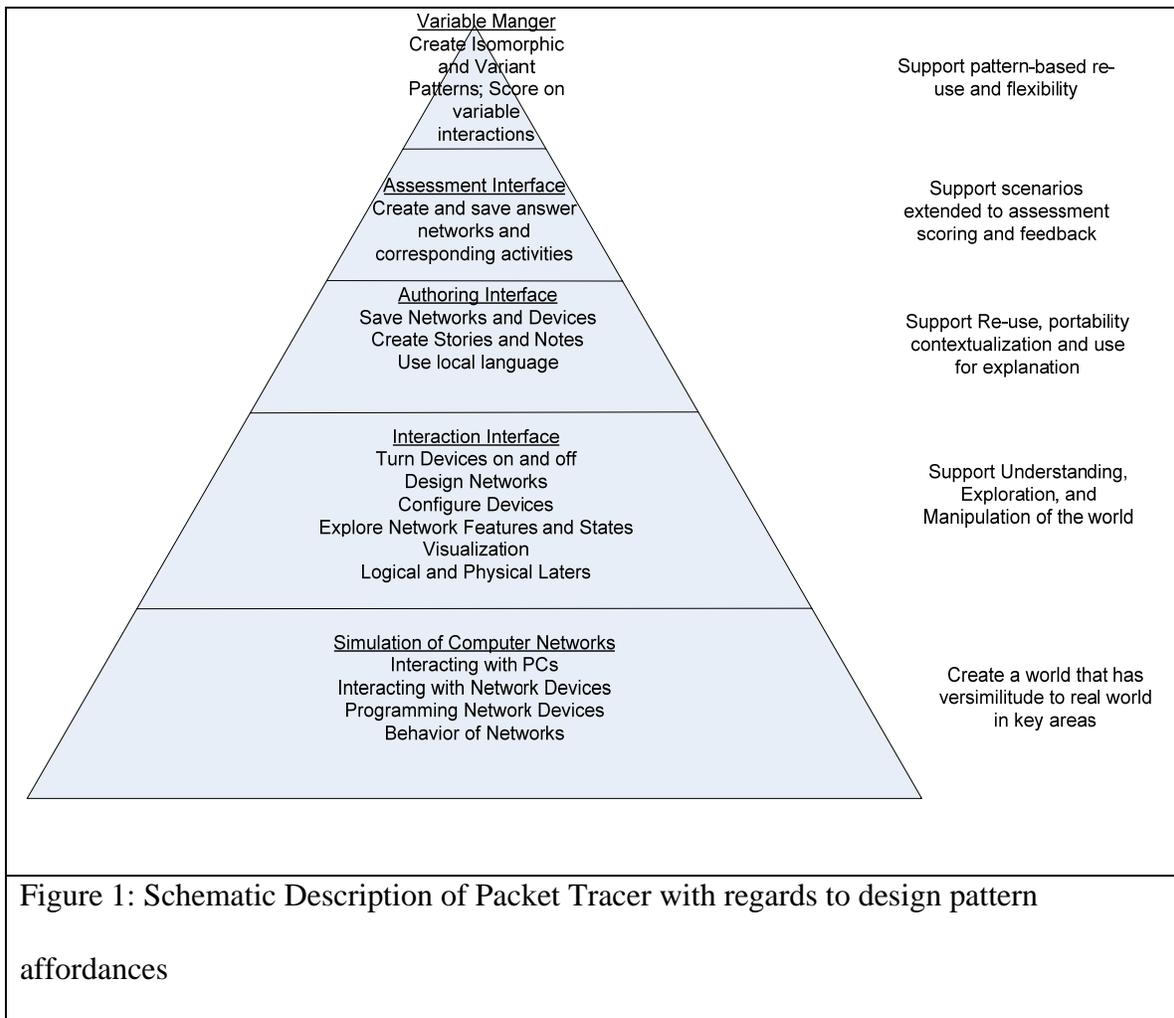
business and social rationale for such work occurring in the context of corporate social responsibility.

### **CORE FEATURES OF THE SOFTWARE ENVIRONMENT**

Over the last 10 years, instructors in the Networking Academy program have come to rely on a 4-way combination of using on-line curriculum, on-line assessments and grade book, hands-on equipment and activities as well as supplemental in class activities. While these instructional supports are generally highly regarded (as indicated by adoption rates and student satisfaction data), they failed to completely address a number of difficulties that perturb such a global program. First, while academies are required to have a hands-on component including the presence of networking equipment in the classroom, the required minimum amount of equipment is designed to give students experience with the equipment, but can not necessarily provide the desired amount of practice if the class is large or if only the minimum amount of equipment is available. Second, even when sufficient amounts of equipment are present to provided the desired amount of hands-on practice, the networks will typically not be of the desired complexity and the available networks of desired complexity are typically (and wisely) off-limits to students because of the possibility of damage or disruption to so-called production networks. The distinction is made between “production” networks (such as for delivery of courses to a classroom or library) and experimental networks on which networking students are free to manipulate equipment. Third, even if complex networks could be made available broadly, there are numerous aspects of the network that consist

of abstract digital entities that are difficult to understand without additional cognitive supports.

The Packet Tracer® software tool was developed to address these needs of scalability of hardware experience, flexibility of hardware experience, and visualization support for learning the abstract concepts and concrete skills of designing, configuring and maintaining data networks. With regard to the design elements that are the focus of the current discussion, Figure 1 provides a schematic depiction of the Packet Tracer application. The breadth of the diagram at any level provides a relative indication of the specificity of the features with regard to supporting design pattern structure in the application. First, we consider the lower two layers that are most connected with the core experience of the student. In the next section, we focus on the subsequent three layers that are more widely used by instructors. This figure makes clear that incorporating a simulation environment into the Networking Academy program requires far more than simply the simulation environment. The supporting layers above the simulation per se help students (and instructors!) work in the environment in ways that develop the patterns of acting and ways of thinking that characterize network engineers—in Shaffer's (2007) phrase, the epistemic frame of the domain.



### Level 1: Simulation of Computer Networking Logic

At its most basic level, Packet Tracer is sometimes characterized as “network simulation software”. While we consider this a fundamental aspect of the system on which all subsequent logic builds, we do not consider it the solitary focus of the application. The computing machinery focused on this level is concerned with the fidelity of simulation of the computer network. Networks consist of hardware devices and software that employ protocols and operating systems to allow devices to communicate. There are standard logics for the ways devices communicate (protocols)

and it is important to simulate with fidelity appropriate to the instructional goals. A key protocol for the Internet is the Internet Protocol (IP); the message unit – in which most data (including text, graphics, voice, and video)\_is transmitted across the internet – is called the packet. Packet Tracing refers to the practice of examining individual packets as they traverse a network. In the case of Packet Tracer, the visualization of detailed networking traffic (packets, represented as envelopes) requires a correspondingly detailed simulation of how the packets change as they traverse the network and the how the states of the devices change during these interactions (for example, how the routing table is updated to account for new devices on the network.

Here it is likewise important to decide where simulation fidelity can be short-cut so as to avoid ongoing duplication of the expensive implementation of real devices. At this level the specific networking protocols and devices must be architected. Packet Tracer® simulates the vast majority of protocols and devices associated with the goals of the Cisco Certified Networking Associated ® exam and corresponding entry level skills. This includes a number of virtual PCs, servers, routers, switches, hubs, IP phones, wireless devices and related devices, connected through various types of virtual cables, that students need repeated practice with in numerous configurations.

## **Level 2: Graphical User Interface and Interaction Layer**

No matter how sophisticated or impressive the simulation level is, it remains an engineering novelty until it is linked to an easily interpretable and navigable user interface. Because Packet Tracer relies on a very rich domain model with elements that do not generalize across other domains, the interface of Packet Tracer is unique in many

ways. The interface design was created by trying to adhere to general Human Computer Interaction design standards, refinement by Use Case analysis and validation many hundreds of hours of student observation. Nevertheless, the uniqueness of some of the representation can become both an impediment and a leverage point for learning. As an impediment, unique features may cause interruption of other tasks and bring in construct irrelevant variance as users apply short term memory processes to figure out interface meaning at the cost of other activities. Alternately, the richness of the interface provides numerous cues by which to engage the student to question “what does that mean” and “why is it like that.”

The central representation of Packet Tracer is the Network Diagram, most commonly called the logical topology. A network diagram is a common and flexible representation that is used in communicating about Networks the composition of elements in a network and the relationship between the elements. Packet Tracer builds off this underlying representation by allowing the diagram to have either a primarily schematic emphasis (a logical topology) or a physical emphasis (a physical topology), and to portray simulated activity in the network in either a Realtime (immediate) mode or a Simulation (protocol event-driven, time-controllable) mode. An example of the Packet Tracer interface in the Logical-Real Time mode is shown in Figure 2.

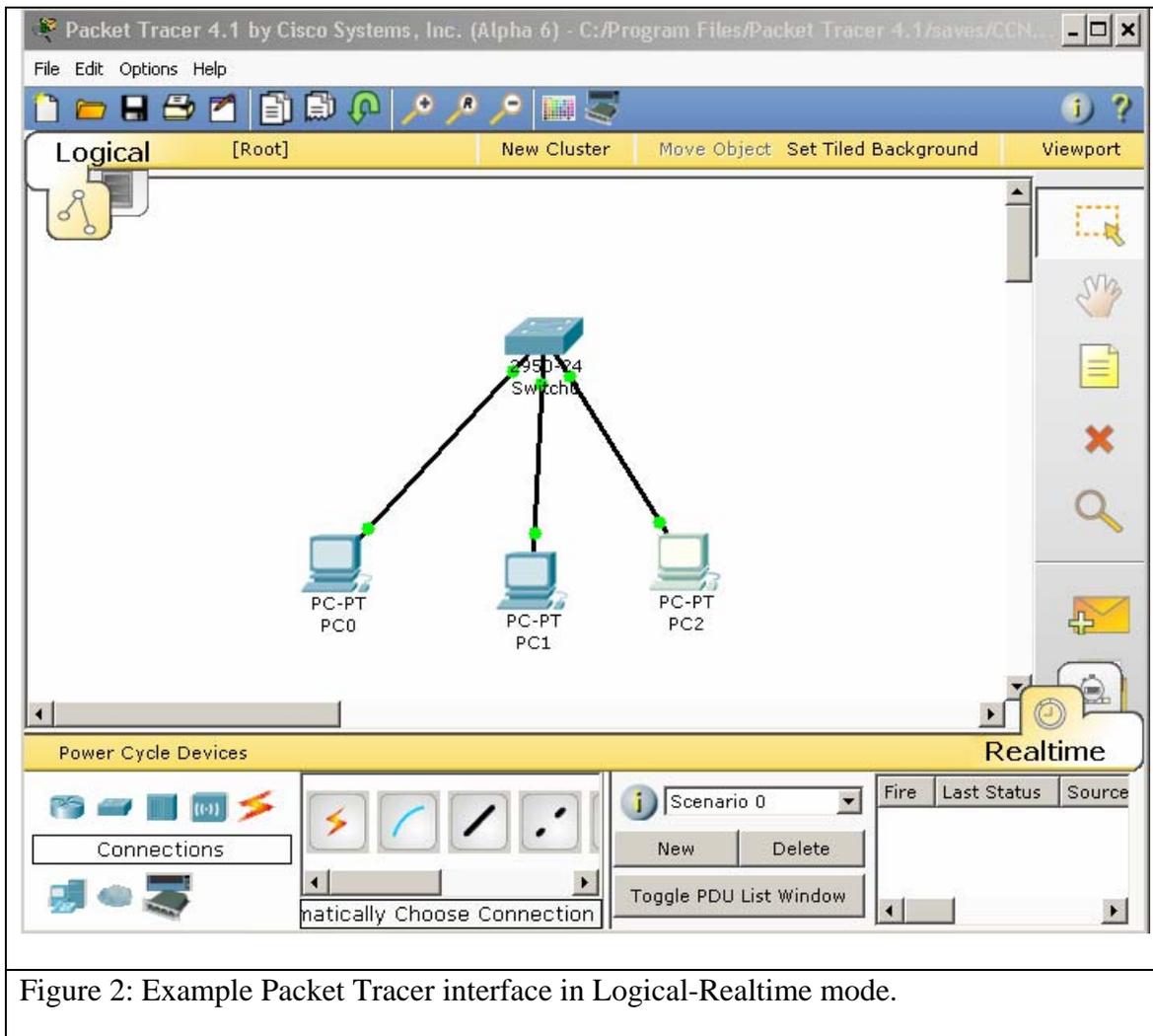


Figure 2: Example Packet Tracer interface in Logical-Realtime mode.

The center of the display holds the network diagram which can either be authored by dragging items from the device and connection palettes in the lower left, or by opening previously saved files from the file dialog. An alternate representation of a network diagram can be created by moving to the Physical network representation which is accomplished by clicking on the tab in the upper left corner of the diagram space. In this view, the same devices placed in the logical representation are available for placement in physical analogs as seen in Figure 3.

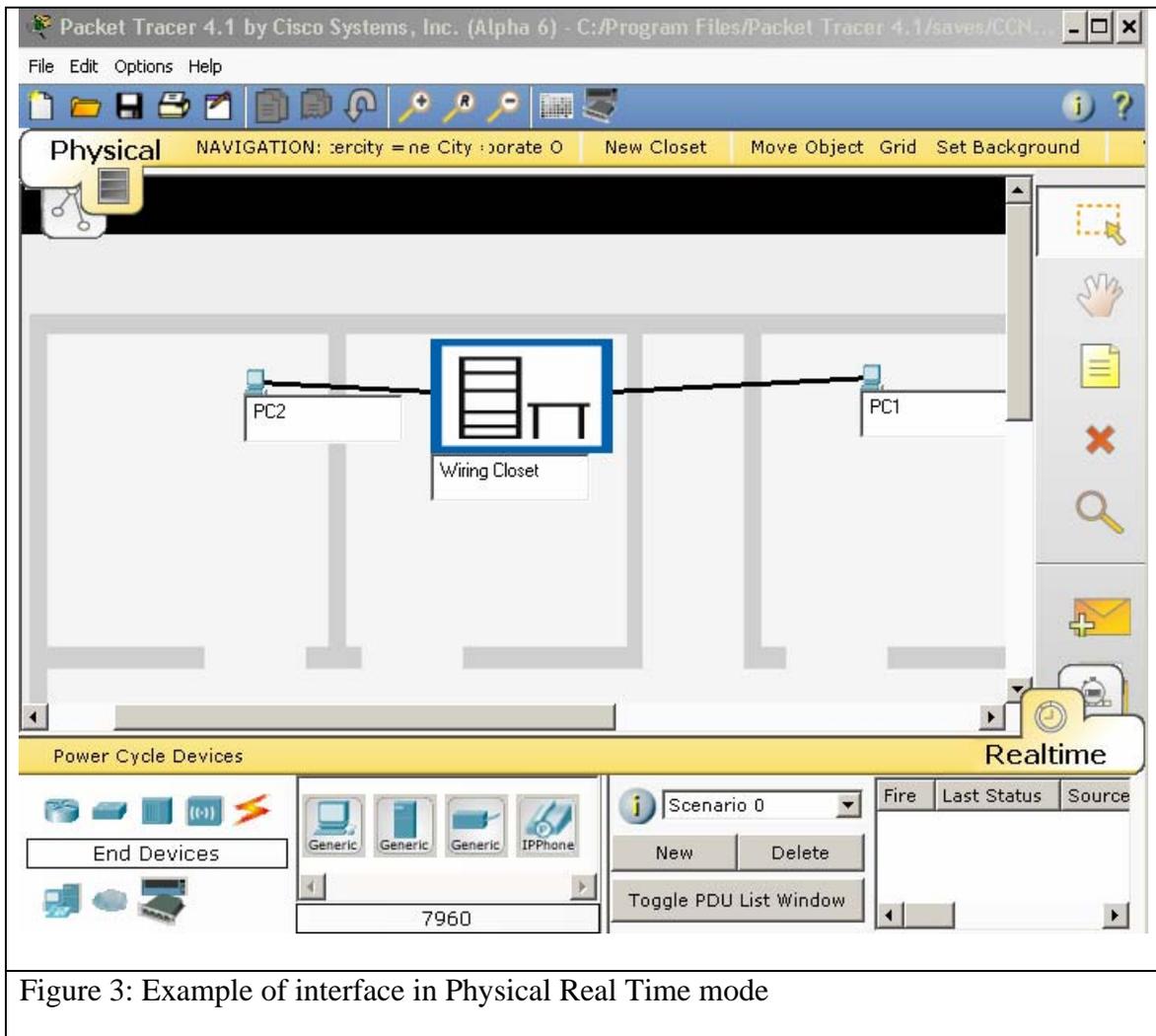


Figure 3: Example of interface in Physical Real Time mode

In either the logical or physical view, the attributes of the devices can be made salient by double clicking on the icon of the device. When this is done, a device specific window is opened with three tabs allowing customization of the physical, networking configuration, or desktop software features of the device. The information presented is device specific though there are some common design features regardless of device.

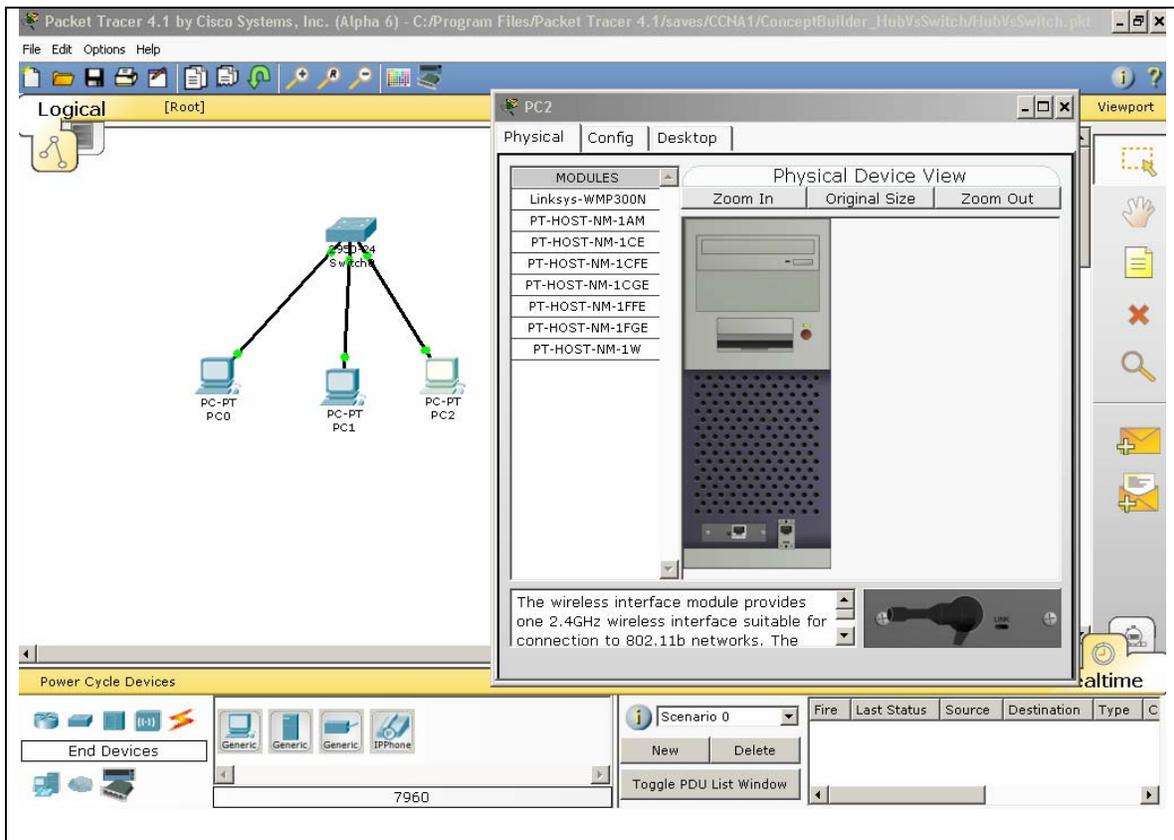


Figure 4. Clicking on the PC icon in the network diagram launches the PC details window.

Consider for example the dialog presented on the right side of Figure 4. In this figure we see the Physical tab presented. On the left side of the picture is a list of possible interface cards that can be placed in the PC. The bottom of the dialog shows a picture of the currently active choice, a wireless antenna. To add it to the PC we would have to first remove the already implemented Ethernet interface shown at the bottom of the PC. Before we remove it however, we would need to notice the red light above the power button that indicates the PC is on. Packet Tracer will not let you remove the card without first turning the PC off. An attempt to do so would lead to a dialog box reminding you of the

importance of this. The remaining tabs in the device detail provide interface to the PC such as a DOS-like command window, a wireless configuration dialog, and simulated informational interfaces required to simulate the full activity and set of representations that exist in a PC to access a data network.

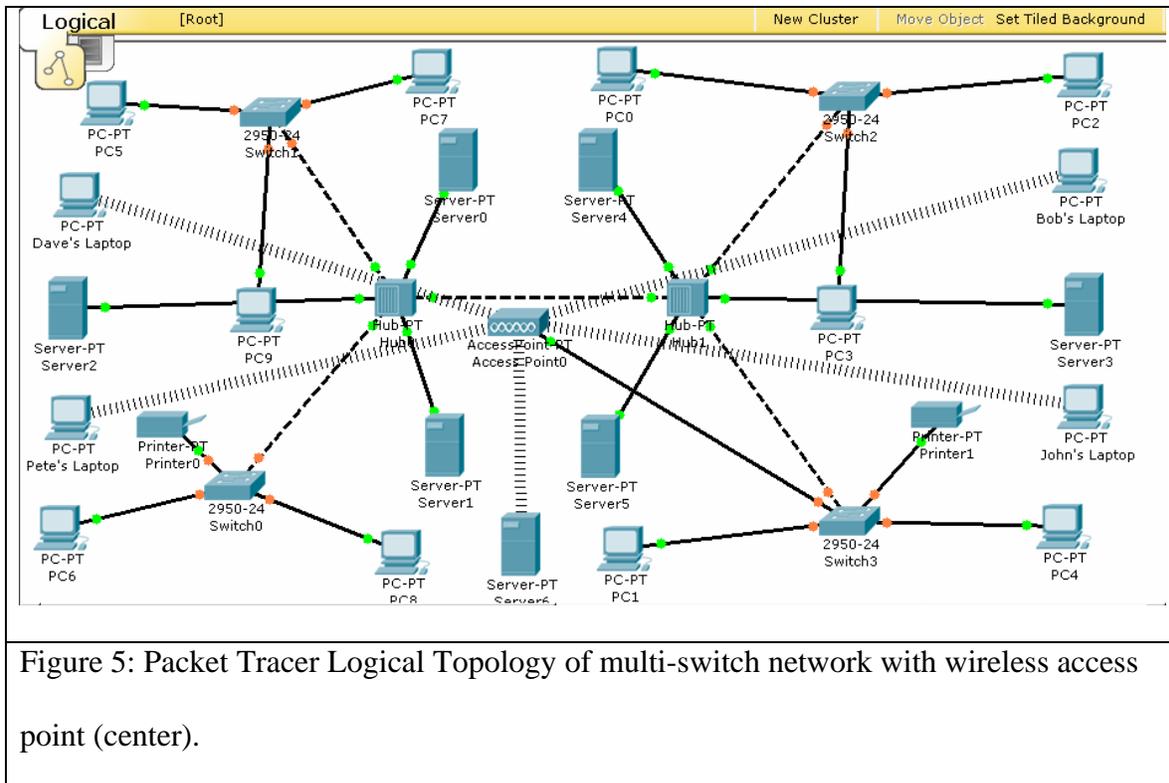


Figure 5: Packet Tracer Logical Topology of multi-switch network with wireless access point (center).

When in the logical mode of a network, a user can configure devices and connections, test for functionality, traverse and troubleshoot the network very much as if they were working on live production network. Typically, after the first few months of training in a job context, much of the work occurs in telnet windows of PCs remotely connected to switches and routers. Accordingly, the simulation windows of Packet Tracer provide excellent fidelity to the on-the-job experience. The reader may recall that this desire to provide high fidelity simulation of complex networks to many students

simultaneously was a central motivation for creating the software system. To give a sense for the possibilities in the program, Figure 5 illustrates an example network used to illustrate a broad range of concepts. Each device on this network needs to be appropriately configured for the network to operate appropriately.

The features described above illustrate the network functionality required for practice in the core areas of design, configuration, troubleshooting and maintenance, in simple and complex environments. Additional features exist to support the understanding of the otherwise unobservable actions occurring in the network that reveal the logic of the hardware and software involved.

Figure 6 is a screen shot of two small networks and the representation of the time-based events that are occurring on the devices. The right side of Figure 6 includes an event list that is configurable by the user. As the various devices send and receive messages the network traffic can be observed and the history of the network can be stopped, reversed and explored. Network traffic is comprised of packets of various protocols, some carrying user data (text, graphics, voice, video) and others carrying networking control information (routing and switching protocols).

By clicking on the event list entry, additional detail regarding the packet involved can be examined. An example is presented in Figure 7. This is an important representation insofar as it presents information, from “inside the packet,” that is typically more difficult to obtain from an active network of real equipment while at the same time displayed in a format consistent with textbooks and on-line curricula. Because the simulated micro-world is open for visual inspection, new dialogs are possible between student and teachers and among students themselves and have been observed

and analyzed from the perspectives of activity theory and computer supported collaborative learning (CSCL) (Frezzo, 2007).

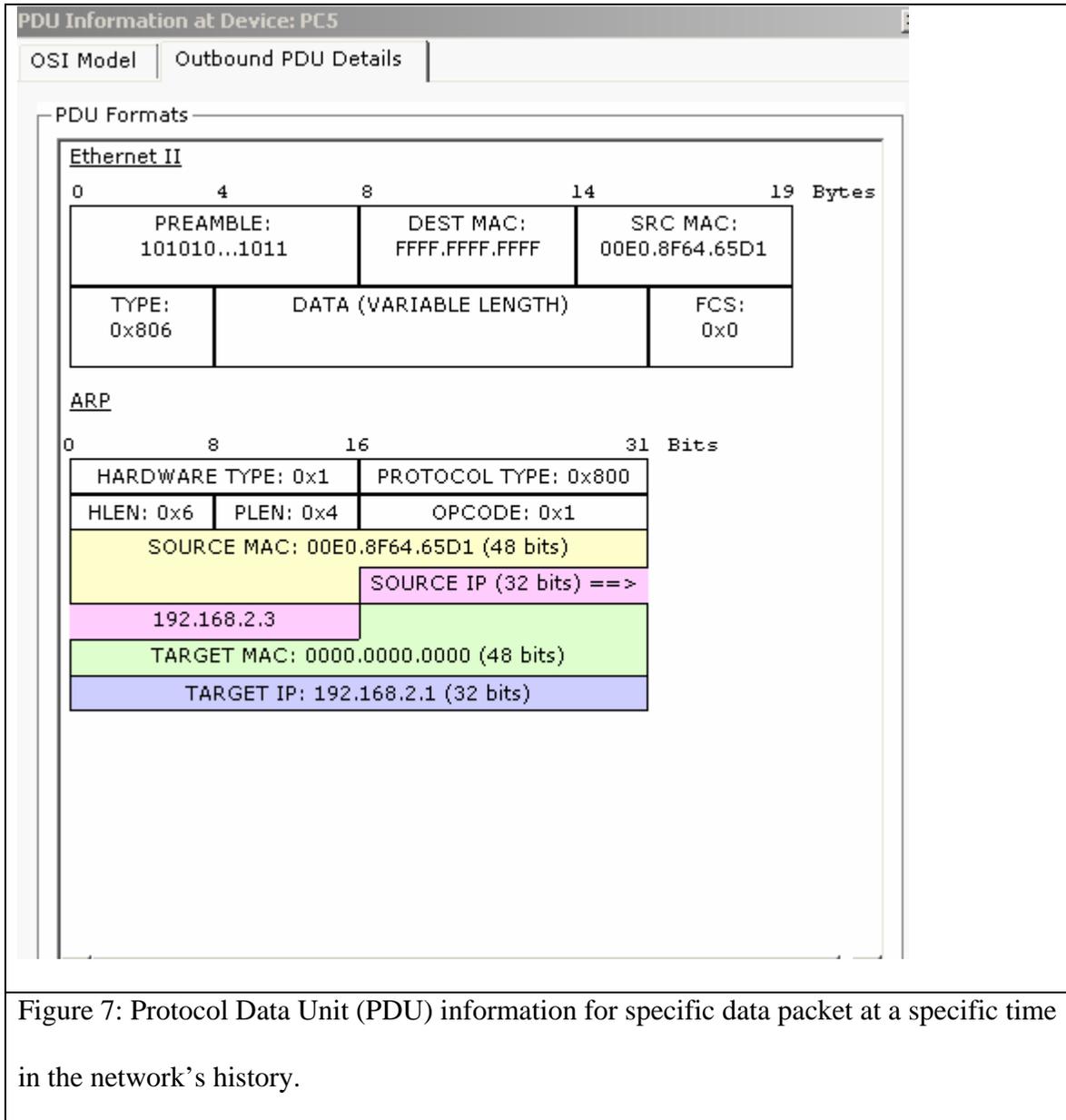
In summary, Packet Tracer provides a broad range of representations and information that support practice in the activities of network design, configuration and troubleshooting, as well as providing otherwise unavailable illustrative and explanatory data about the inner workings of the complex system. The combination of these two major aspects of the tool provides the groundwork for both directed and undirected learning advances.

The screenshot displays the Packet Tracer interface. The main window shows a network topology with a switch and a hub, each connected to three PCs. The event list on the right shows the following data:

Vis.	Time (s)	Last Device	At Device	Type	Info
<input type="checkbox"/>	0.000	--	PC1	ICMP	
<input type="checkbox"/>	0.000	--	PC1	ARP	
<input type="checkbox"/>	0.000	--	PC3	ICMP	
<input type="checkbox"/>	0.000	--	PC3	ARP	
<input type="checkbox"/>	0.000	--	PC5	ICMP	
<input type="checkbox"/>	0.000	--	PC5	ARP	

The simulation controls at the bottom include 'Power Cycle Devices', 'PLAY CONTROLS' (Back, Auto Capture / Play, Capture / Forward), and 'Event List' (Scenario 0, New, Delete, Toggle PDU List Window). The event list filters show 'Visible Events: ARP, ICMP, VTP, STP, OSPF, DTP, TELNET, TFTP, HTTP, DNS'.

Figure 6. Packet Tracer interface for viewing time-based events, with different IP packets represented as envelopes.



## DESIGN PATTERNS IN THE TOOL FEATURES AND USE

The tools described above constitute authoring tools for networking microworlds. An author can create a network and set tasks for others to complete. These tasks can be

challenging and motivating in a manner that elicits the motivation of a test or a game (Behrens, Frezzo, Mislevy, Kroopnik & Wise, 2007). Alternately, students can set about on undirected exploration to find valuable realization or commonplace confusion. Given these broad range of possibilities and the bewildering array of academic, employment and pedagogical backgrounds instructors have around the world, we are faced with the challenge of promoting flexibility and customization, while at the same time providing tools for structuring understanding, communication, re-use. To accomplish this, we use the concept of patterns at multiple levels in tools use and tool features.

### **Level 3: Authoring Features and application**

The first set of scaffolding we provide is in the form of pre-made networking activities that follow one of four possible patterns: concept builders, skill builders, design challenges and troubleshooting scenarios. Concept builders are activities that focus on illustrating a particular concept and may function either as a demonstration, or require more active problem solving and impasse resolution. Skill builders focus on practice of procedural skills, typically in a more isolated rather than integrated setting. Design challenges are open ended tasks with multiple possible approaches and no single “answer” necessarily. Troubleshooting scenarios also open ended tasks that require understanding a design, identifying faults and generating and testing hypotheses.

Recently Packet Tracer was integrated directly into the courses, and more elaborated activities, based on design patterns, were used. These patterns included three general types: embedded activities (meant for formative assessment within a chapter, which include concept building demonstrations, design, configuration, and troubleshooting), lab analogs (which follow hands-on labs, meant to give pre- and post-

practice before and/or after a student uses real equipment), and skills integration challenges (cumulative across chapters and courses, integrating elements of design, configuration, and troubleshooting).

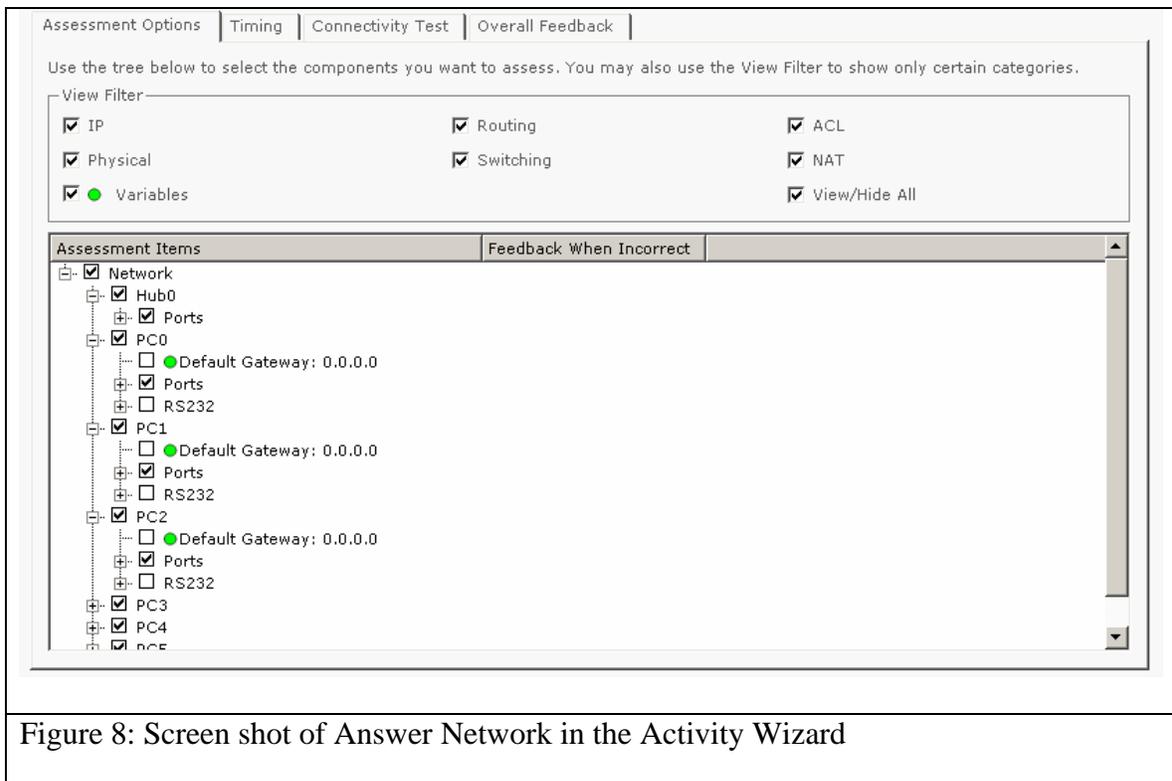
The patterns associated with these activity types serve multiple purposes. First, it helps the development team communicate about the different types of goals appropriate in this domain and open the conversation about the essential elements of each pattern. This supports both intra-team communication as well as management of goal types insofar as it allows us to ask which types are appropriate for which objectives. Second, this nomenclature fosters a dialog with the instructor community because we use this language in our own instructional materials (e.g. built-in help) and include files of each type for most parts of the curriculum. We have found this type of organizational function to be extremely valuable in that it moves the global community toward a common language which can improve communication.

While the activity patterns used at this stage are designed, in part, to promote common conceptualization, the tool itself allows for great individualization. Among other features, Packet Tracer allows authoring of stories, notes, graphics, web pages, and information windows and menus in local languages including double-byte and right to left Semitic languages.

#### **Level 4: Assessment Authoring**

Because we believe in a comprehensive notion of instruction that integrates curriculum and assessment, it was a natural evolution of our work to build in assessment authoring features into Packet Tracer. The general strategy for this is the specification of (1) scaffolding, in the form of instructions, timers, an initial network, and GUI-locking

tree, that serves as the presentational features of the assessment as described in the ECD literature (Mislevy , Steinberg & Almond, 2003), and an (2) “answer network” with “grading tree” that serves as the evidence identification rules over the range of observables to be considered in task level scoring. Specification of these elements is accomplished by created a specific network, and then refining a corresponding detailed tree that allows the author to indicate which features are to be considered and which features are deemed irrelevant to the construct under examination. Possible features for comparison are automatically generated by an “Activity Wizard” that converts the logical topology into a list of relevant possibilities. Figure 8 is a screen shot of an answer network illustrating nodes that are logical possibilities (they appear at all) and those that are elements for comparison with the work product (they have check marks).



This affordance is a significant step in the enablement of instructors to construct skills-based automatically scored assessments. The highly structured format of the interface helps instructors consider the structure of their assessments and what elements are most salient to the evidence they are trying to collect. Accordingly, not only does the interface have to function in a manner that supports these views, but additional training and support is likely needed.

Using these tools, patterns of assessment activity can easily be created. For example, a troubleshooting activity often consists of a well-designed network that has one or more bugs introduced into it. This can easily be handled by creating a working network, save it, and then retrieve it and save with a bug imbedded. By change the location or type of fault across multiple files, one can create numerous isomorphs or variants with relatively small amount of work.

### **Level 5: Assessment authoring with dynamic variables**

The example given above of creating isomorphic items presupposes a template or design pattern for the task and an identification of which elements should vary and which should not. After that, we have described a manual process for introducing variation. To help our instructor community apply such logic, Packet Tracer now allows the creation of initial networks and answer trees that allow for selection of variable values from variable string pools or numeric values.

By combining different aspects of these features, instructors (and the development team) will be able to create large pools of variant items. For example, problem stories may be made to vary by manipulating elements of the story. The logical values of

elements of the networks could likewise be manipulated so that each student had a different network in a possibly undetectable manner. And answer networks will not have to look for identical matches of device features (e.g. match to one single network address), but could generalize the rule to variables that match the relevant features, for example router names or IP addresses.

Again, we consider this underlying structure and its distribution to instructors a major improvement and value to the instructor community. We hope that by providing these structures, instructors will be put in a place to question their instructional an assessment practice and follow and create design patterns that can be managed through such templates. Given the global and network base of the community, we hope to see such templates and patterns shared around the globe.

## **SUMMARY**

Packet Tracer is a complex tool that works at several levels. This paper described how those different levels build one upon the other to allow the individual authoring of networking micro-worlds for instructional and assessment uses. By using a unified architecture, authoring, and variable management, we can promote pattern-based authoring and re-use to gain efficiency and improve communication about the patterns that drive educational structure in our domain.

**REFERENCES**

- Behrens, J. T., Collison, T. A., & Demark, S. F. (2005) The Seven Cs of comprehensive assessment: Lessons learned from 40 million classroom exams in the Cisco Networking Academy Program. In S. Howell and M. Hricko (Eds.), *Online Assessment and Measurement: Case Studies in Higher Education, K-12 and Corporate* (pp 229-245). Hershey, PA: Information Science Publishing.
- Behrens, J. T., Frezzo, D. C., Mislevy, R. J. Kroopnik, & Wise, D. (2007). Structural, Functional and Semiotic Symmetries in simulation based games, and assessments. In E. L. Baker, J. Dickieson, W. Wulfeck, & H. F. O'Neil (Eds.) *Assessment of Problem Solving Using Simulations*. Mahwah, NJ: Erlbaum.
- Behrens, J. T, Mislevy, R. J., Bauer, M., Williamson, D. W., & Levy, R. (2004). Introduction to Evidence Centered Design and lessons learned from its application in a global e-learning program. *International Journal of Testing*, 4, 295-301.
- Frezzo, D. C. (2007). Using Activity Theory to Understand the Role of a Simulation-Based Learning Environment in a Computer Networking Course. Unpublished doctoral dissertation, University of Hawai'i, Manoa.
- Mislevy, R.J., Steinberg, L.S., & Almond, R.G. (2003). On the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives*, 1, 3-62.
- Murnane, R.J. , Sharkey, N.S. & Levy, F. (2004). A role for the internet in American education? Lessons from the Cisco Networking Academies. In P. A. Graham & N. G. Stacey (Eds.), *The Knowledge Economy and Postsecondary Education: Report of a Workshop* (pp.127-158). Committee on the Impact of the Changing Economy on the

Education System, National Research Council. Washington, D.C.: National Academies Press.

Porter, M. E., & Kramer, M. R. (2002). The competitive advantage of corporate philanthropy. *Harvard Business Review*, 80, 12, 56-68.

Shaffer, D.W. (2007). *How computer games help children learn*. New York: Palgrave Macmillan.