

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

Dennis C. Frezzo · John T. Behrens ·
Robert J. Mislevy

Published online: 26 September 2009
© The Author(s) 2009. This article is published with open access at Springerlink.com

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 Academies with a simulation tool for computer networks called Packet Tracer. The focus is on a conceptual support framework for instructors in over 9,000 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 network

simulator that supports the use of Packet Tracer in the distributed community of instructors and students.

Keywords Assessment · Design patterns · Packet Tracer · Simulation

Introduction

The Cisco Networking Academy program is a collaborative endeavor including over 9,000 schools in 160 countries aimed at supporting the learning of basic and intermediate level computer networking skills. To help accomplish this goal, Cisco provides a toolbox of aides to the instructors and students including on-line interactive curricula, on-line assessments for chapter, mid-term and course-final feedback with an integrated on-line grade book for instructors and students, a network of cooperating educational institutions that train instructors in partnering institutions, discounted equipment to provide hands-on instruction and assessment, global support desk for instructors and other features.

While widely adopted, 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 the provision of feedback from fixed-response tasks, we want to provide the learners and instructors with tools that enable them 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.

D. C. Frezzo · J. T. Behrens
Cisco, Networking Academy Learning Systems Development,
SJC05/3/4 325 East Tasman Drive, San Jose, CA 95134, USA

D. C. Frezzo
e-mail: dfrezzo@cisco.com

J. T. Behrens
Department of Psychology, University of Notre Dame, Notre
Dame, IN, USA
e-mail: jbehrens@cisco.com

R. J. Mislevy (✉)
Department of Measurement, Statistics and Evaluation,
University of Maryland, EDMS, Benjamin 1230-C, College
Park, MD 20742, USA
e-mail: rmislevy@umd.edu

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 to use. 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 authoring, and creation, tool that promotes instructional enablement and local customization. This is accomplished by considering cognitive interactions with the domain, and promoting design patterns in both the creation of scenarios and the development of authoring tools that promote pattern based thinking. This is discussed 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 summary and discussion of limitations.

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, especially in underserved communities. 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 campuses and the Internet. The program is a cooperative undertaking between Cisco, the Cisco Learning Institute, and approximately 9,000 public educational institutions across more than 160 countries. At present, more than 75% of participating academies are located outside of the United States. Cisco 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 75 million on-line exams across 11 languages in the last 10 years. The structure and distinguishing features of the assessment program have been described by Behrens et al. (2005) with additional information provided by Behrens et al. (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 hands-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 et al. (2004) provide a discussion of the origin of the program, while Levy and Murnane (2004) describe issues related to technological application of the curriculum and assessment. Porter and 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 (1) on-line curriculum, (2) on-line assessments with grade book, (3) hands-on equipment and activities, and (4) 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 can not always provide the desired amount of practice if the class is large or the students require extended practice for lack of outside experience. Second, even when sufficient amounts of equipment are present to provide the desired amount of hands-on practice for skill acquisition, the networks used will typically not be able to cover the scope of desired complexity. One might describe this situation as a high student-to-network ratio. Furthermore, the available physical 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. That is to say, the electronic world behind the machines is complex in a way not easily understood by working with physical equipment alone. The physical equipment can help refine and automate procedural skills, but has few affordances for promoting mental models of the underlying communication activity.

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, and support authoring and deployment of complex skill assessment. With regard to the design elements that are the focus of the current discussion, Fig. 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 subsequent section, we focus on the next three layers that are more widely used by instructors. This figure reflects the idea that incorporating a simulation environment into the Cisco Networking Academies requires far more than simply a network simulator. The supporting layers above the network simulation 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.

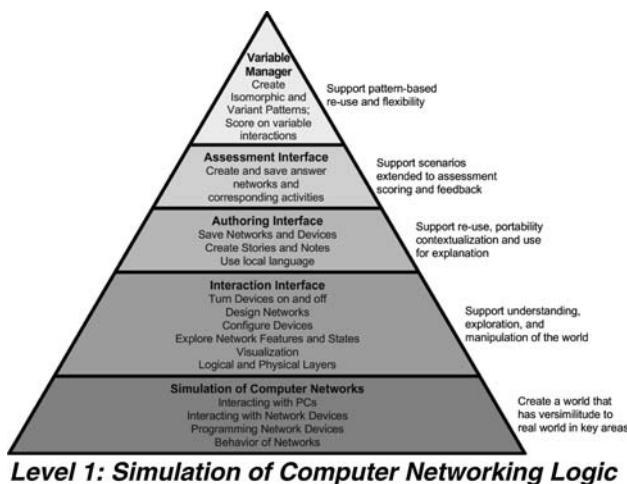


Fig. 1 Schematic description of Packet Tracer with regards to design pattern affordances

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, operating systems, and application software that employ protocols to allow devices to communicate. There are standard logics for the ways devices communicate, such as the TCP/IP protocols. Because perfect comprehensive simulation is not possible, 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 software, 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, by which a routing device determines how to forward a packet towards its destination, is updated to account for new devices on the network.

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 designed with great detail to provide an experience that closely resembles the use of real equipment. Packet Tracer[®] simulates the vast majority of protocols and devices associated with the goals of entry level skills taught in the Cisco Networking Academies, and tested via the Cisco Certified Networking Associate[®] exam. 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 hours of student observation (Frezzo 2009). Nevertheless, the uniqueness of some of the representations 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 understand 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.” As learners move from novices to experts and interaction with the interface becomes more automatic, we expect that the interface becomes less of an impediment and more of an advantage.

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 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, visualization) mode. An example of the Packet Tracer interface in the Logical-Real Time mode is shown in Fig. 2.

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, authored by instructional designers, instructors, or other students, 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 such as a map of a building seen in Fig. 3.

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 features of the device, GUI shortcuts for configuring the device, or access to simulated command-line interface (CLI) configuration of the devices. The information presented is device specific though there are some common design features regardless of device.

Consider for example the dialog presented on the right side of Fig. 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 personal computer (PC). The bottom of the dialog shows a picture of the currently active choice, a wireless antenna. To add the antenna to the PC we would have to first remove the already implemented Ethernet interface card shown at the

Fig. 2 Example Packet Tracer interface in logical-realtime mode

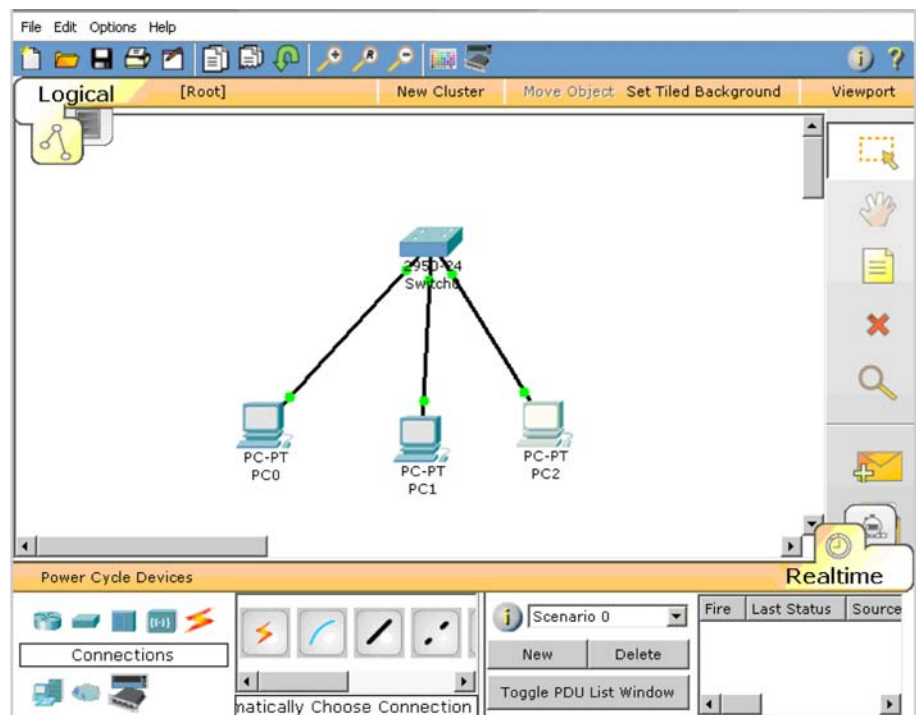


Fig. 3 Example of interface in physical real time mode

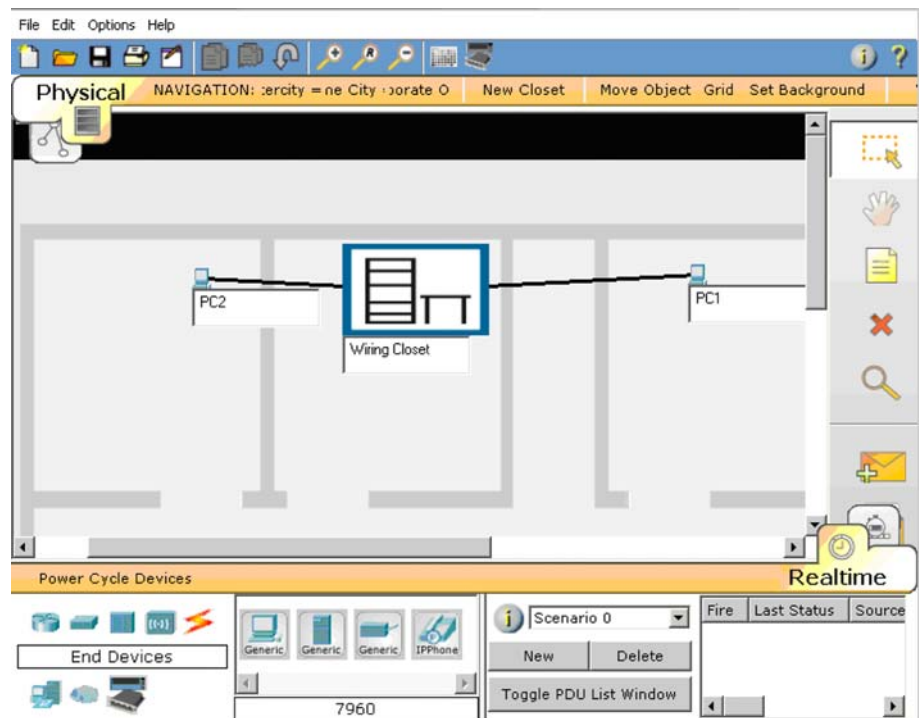
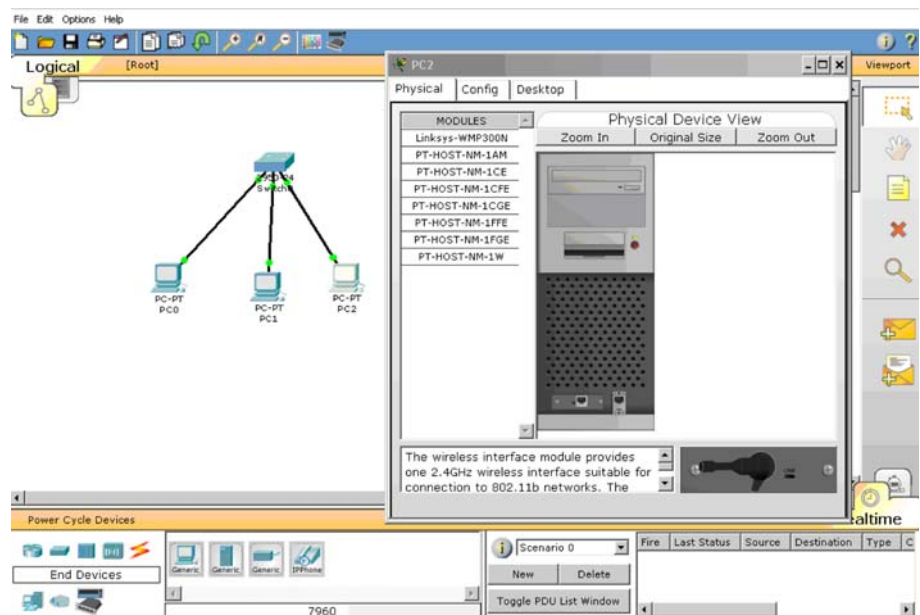


Fig. 4 Clicking on the PC icon in the network diagram launches the PC details window



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 act. The remaining tabs in the device detail provide interface to the PC such as a DOS-like command window, a wireless configuration dialog, and other simulated interfaces required to allow the full activity

and set of representations that are required in a PC to access a data network.

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, most work occurs in telnet windows of PCs remotely connected to switches and routers. Accordingly, the simulation windows

Fig. 5 Packet Tracer logical topology of multi-switch network with wireless access routers

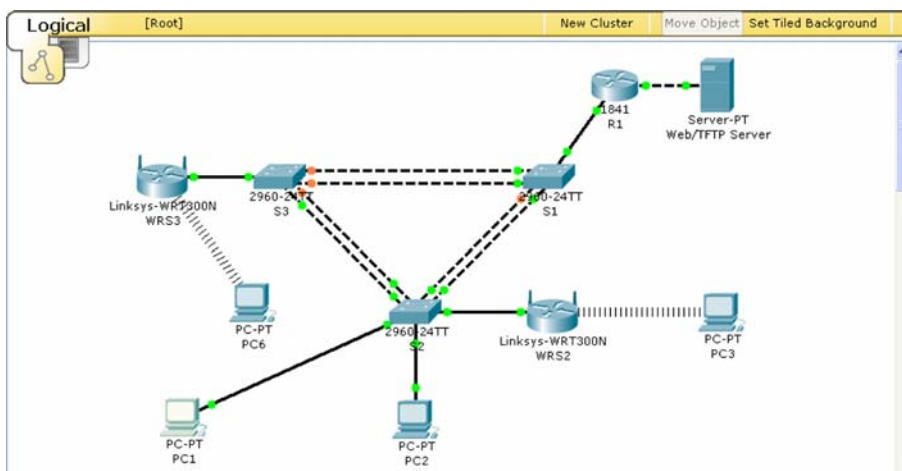


Fig. 6 Packet Tracer interface for viewing time-based events, with different IP packets represented as envelopes

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	

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, Fig. 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.

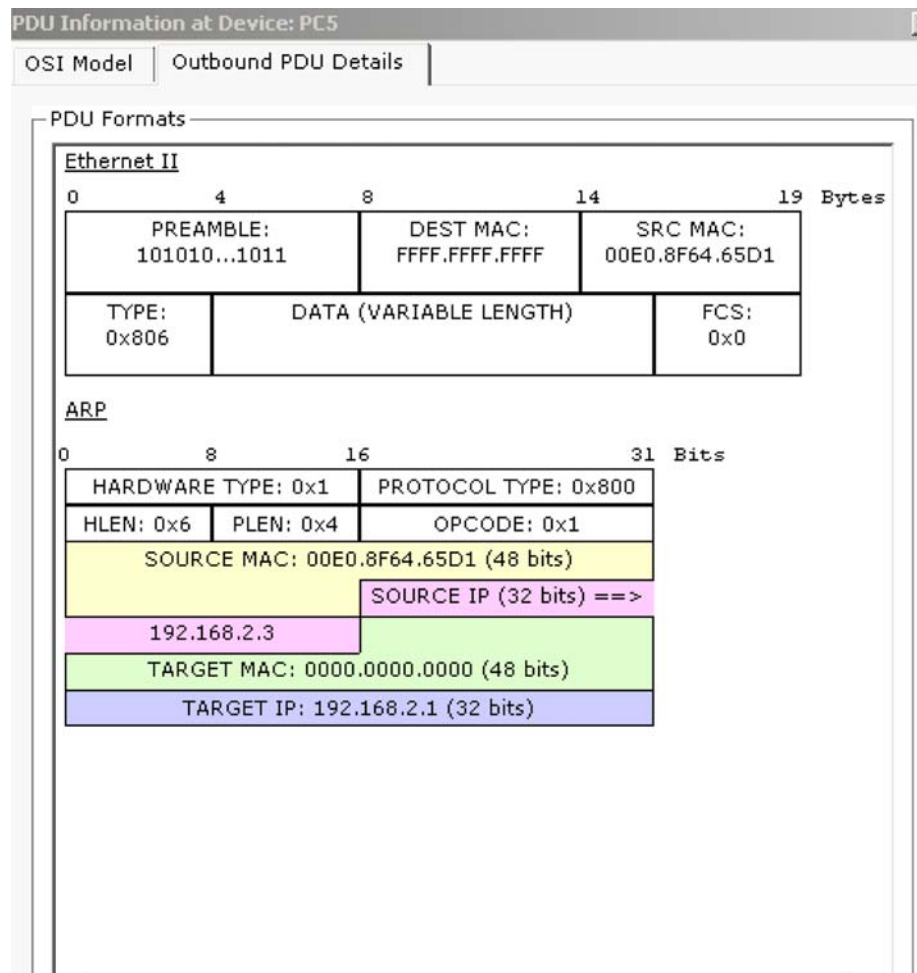
The features described above illustrate the network functionality required for practice in the core areas of design, configuration, troubleshooting and maintenance, in simple and complex environments. Additional features exist to support the understanding of the otherwise unobservable actions occurring 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 Fig. 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 information).

By clicking on the event list entry, additional detail regarding the packet involved can be examined. An example is presented in Fig. 7. This is an important representation insofar as it presents information, from inside the packet, 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. Such expanded dialogs and changing of epistemic frames have been observed and analyzed from the perspectives of

Fig. 7 Protocol data unit (PDU) information for specific data packet at a specific time in the network’s history



activity theory and computer supported collaborative learning (CSCL) by Frezzo (2009).

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 an the increasingly ubiquitous complex system of the computer network. The combination of these two major aspects of the tool provides the groundwork for both directed and undirected learning advances.

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 et al. 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, and 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

One approach to scaffolding we provide is in the form of instructor help resources and pre-made networking activities that follow one of four possible patterns organized around different collections of knowledge, skills and abilities (KSAs): 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 builder activities 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 the possibility of multiple correct approaches. Troubleshooting scenarios are also free-response tasks that require understanding a network design, identifying faults, and generating and testing hypotheses.

In addition to organizing activities according to type of target KSA, Packet Tracer activity patterns may also be understood in terms of variation in the openness directions and specificity of feedback given from tasks. These variations are aligned with different levels of scaffolding required as individuals move from novice to expert in different sub-domains in the curriculum. These patterns include: embedded activities (meant for formative assessment within a chapter, which include concept building demonstrations, design, configuration/skill-building, 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 or options if real equipment is not available), and skills integration challenges (cumulative across chapters and courses, integrating elements of design, configuration/skill-building, 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 of task are appropriate for which stages of learning. 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; instructor training) and include files of each type for most parts of the curriculum. We have found this type of organizational function to be extremely

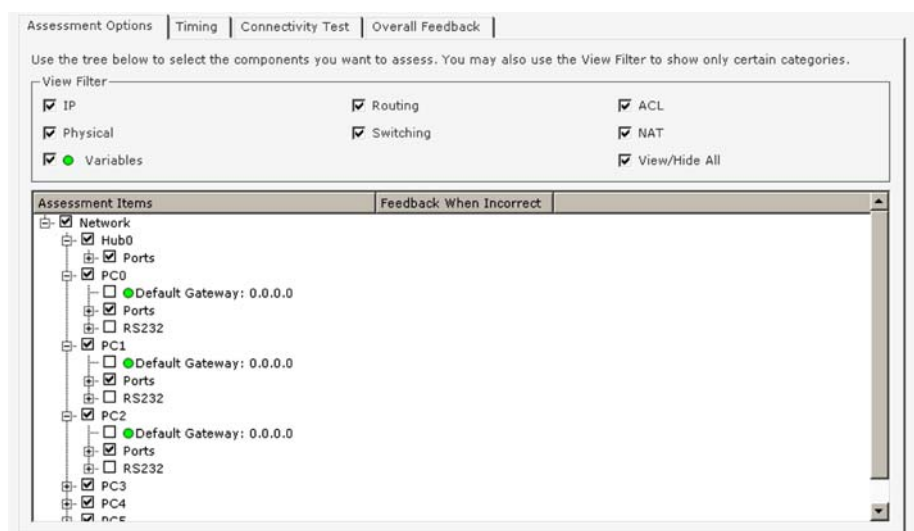
valuable in that it moves the global community toward a common language which can improve communication.

While the activity patterns used at this stage are designed, in part, to promote common conceptualization, the tool itself allows for 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 or Asian 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) end-user micro-world, in the form of formatted 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 et al. 2003), and (2) an “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 creating a specific network, and then refining a corresponding detailed tree that allows the author to indicate which features are to be considered in scoring and which features are deemed irrelevant to the construct under consideration. 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 all appear; they can be seen as a summary of the

Fig. 8 Screen shot of answer network in the Activity Wizard



extent of the domain model) and those that are elements for comparison with the work product (they have check marks).

This affordance is a significant step in enabling 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 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 be handled, with the help of the Activity Wizard, by creating a working network and labeling it the “answer network”, saving it, adding instructions and timers, introducing one or more bugs into the answer network and saving that as the “initial network”, and saving the whole scaffolded “package” as a Packet Tracer *activity* file. Working off a master answer network in this way, numerous complex variants and isomorphs can be established with a 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) are 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. Answer networks do not have to look for identical matches of device features (e.g. match to one single network address), but can generalize the rule to variables that match the relevant features, for example router names or an element in a range of IP addresses.

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 and assessment practice and create design patterns that can be managed through such templates, either through incremental improvements in existing activities or authoring from scratch. 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 design architecture, authoring approach, 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.

Despite this full range of computational advances, a number of limitations remain. First, as the breadth of features in Packet Tracer increase, there is a concomitant need for extended professional development for users. While we are happy to have an extended feature set for central development for which the most advanced features of the software would be used by a small set of pedagogical innovators, the depth of use is clearly a function of professional development in the instructor community. To address this issue the Cisco Networking Academy Technical Advocacy team provides both in-person and on-line meetings to provide instruction and technical support to the instructor community. Detailed tutorial and help files are also included with the software.

As second limitation of the current deployment is the management of the increasingly large pool of Packet Tracer micro-world files and corresponding assessment files. Within the Cisco development team, work is underway to integrate the files with other content management systems. However, the complexity of the tasks and scoring systems possible provides a unique set of opportunities and challenges. To address the related need for support of sharing activity files across the instructor community, a service-based web site to support Web 2.0 collective intelligence interactivity is currently under development and testing.

In sum, the logical framework of Evidence Centered Design (Mislevy et al. 2003) provides a scalable framework to understand and extend the features of Packet Tracer for increasingly sophisticated assessment design and authoring. Nevertheless, these technical advances created additional stresses on the human systems for instructor training and support and content management. Despite these challenges, the models discussed here provide a global community with rich and extensible tools to employ

educational schemes created by others, or to individualize activities to meet their own unique requirements.

Acknowledgments The authors would like to thank the instructors and students around the globe that have contributed to the design and development of Packet Tracer by the use and feedback they provide.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Behrens JT, Mislevy RJ, Bauer M, Williamson DM, Levy R (2004) Introduction to evidence centered design and lessons learned from its application in a global e-learning program. *Int J Test* 4:295–301
- Behrens JT, Collison TA, Demark SF (2005) The seven Cs of comprehensive assessment: lessons learned from 40 million classroom exams in the Cisco Networking Academy Program. In: Howell S, Hricko M (eds) *Online assessment and measurement: case studies in higher education, K-12 and corporate*. Information Science, Hershey, pp 229–245
- Behrens JT, Frezzo DC, Mislevy RJ, Kroopnick M, Wise D (2007) Structural, functional and semiotic symmetries in simulation based games, and assessments. In: Baker EL, Dickieson J, Wulfek W, O'Neil HF (eds) *Assessment of problem solving using simulations*. Erlbaum, Mahwah
- Frezzo DC (2009) 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
- Levy F, Murnane RJ (2004) *The new division of labor: how computers are creating the next job market*. Princeton University Press, Princeton
- Mislevy RJ, Steinberg LS, Almond RG (2003) On the structure of educational assessments. *Meas Interdiscip Res Perspect* 1:3–62
- Murnane RJ, Sharkey NS, Levy F (2004) A role for the internet in American education? Lessons from the Cisco Networking Academies. In: Graham PA, Stacey NG (eds) *The knowledge economy and postsecondary education: report of a workshop*. Committee on the Impact of the Changing Economy on the Education System, National Research Council, National Academies Press, Washington, D.C., pp 127–158
- Porter ME, Kramer MR (2002) The competitive advantage of corporate philanthropy. *Harv Bus Rev* 80(12):56–68
- Shaffer DW (2007) *How computer games help children learn*. Palgrave Macmillan, New York