The Pedagogical Design Studio: Exploiting Artifact-Based Task Models for Constructivist Learning

James C. Lester
Dept. of Computer Science
North Carolina State University
Raleigh, NC 27695-8206
(919) 515-7534
fax: (919) 515-7896
lester@adm.csc.ncsu.edu

Patrick J. FitzGerald
Dept. of Design & Technology
North Carolina State University
Raleigh, NC 27695-8206
(919) 515-8364
fax: (919) 515-7330
pat_fitzgerald@ncsu.edu

Brian A. Stone
Dept. of Computer Science
North Carolina State University
Raleigh, NC 27695-8206
(919) 515-7134
fax: (919) 515-7896
bastone@eos.ncsu.edu

ABSTRACT

Intelligent learning environments that support constructivism should provide active learning experiences that are customized for individual learners. To do so, they must determine learner intent and detect misconceptions, and this diagnosis must be performed as non-invasively as possible. To this end, we propose the *pedagogical design studio*, a design-centered framework for learning environment interfaces. Pedagogical design studios provide learners with a rich, direct manipulation design experience. By exploiting an *artifact-based task model* that preserves a tight mapping between the interface state and design sub-tasks, they non-invasively infer learners' intent and detect misconceptions. The task model is then used to tailor problem presentation, produce a customized musical score, and modulate problem-solving intervention. To explore these notions, we have implemented a pedagogical design studio for a constructivist learning environment that provides instruction to middle school students about botanical anatomy and physiology. Evaluations suggest that the design studio framework constitutes an effective approach to interfaces that support constructivist learning.

Keywords

Learning environments, tutoring systems, design, task models.

INTRODUCTION

Constructivist learning has received increasing attention in the education community in recent years. Because of its emphasis on the active role played by the learner as he or she acquires new concepts and procedures [19], constructivism has made considerable gains relative to more purely didactic approaches. A particularly intriguing form of the constructivist's learning-by-doing techniques might be termed "learning-by-designing." In the process of designing an artifact, learners—by necessity—come to understand the rich interconnections between the artifacts they devise and the environmental constraints that determine whether a given design will meet with success.

To most effectively support learning-by-designing, learning environments should provide customized design experiences that are tailored to the problem-solving history of the current learner. However, customization techniques such as individualizing problem presentation and determining when to intervene with assistance require the system to maintain an up-to-date task model that reflects the learner's intentions and detects misconceptions. Unfortunately, because direct probing of the learner would interfere with a purely constructivist experience, design-centered learning environments call for an approach to diagnosis that is much less invasive and disruptive. While providing advice for design tasks is a much investigated topic [9] and efforts have been made to study how to automate the instruction of design per se [10], the primary contributions of other design-centered learning environments are in theories of remindings [8] and constraint negotiation [21].

To address the diagnostic issues of constructivist learning environments, we propose the *pedagogical design studio*, an interface framework for learning-by-designing that supports customized constructivist learning (Figure 1). In this framework, we first devise a design task that is componentwise segmented, i.e., it is decomposed into sub-tasks, each of which corresponds to making design decisions about a particular type of artifactual component. We then use this segmentation to develop an interface whose functionalities are tightly coupled to an *artifact-based task model*. Finally, we show how the task model for an individual learner can be used to dynamically present customized problems, produce a contextualized musical score, and decide when and how to intervene with problem-solving advice.

In this paper, we first outline the goals of constructivist learning environments and discuss their interface design implications. We then introduce pedagogical design studios and their artifact-based task models. After describing how in-

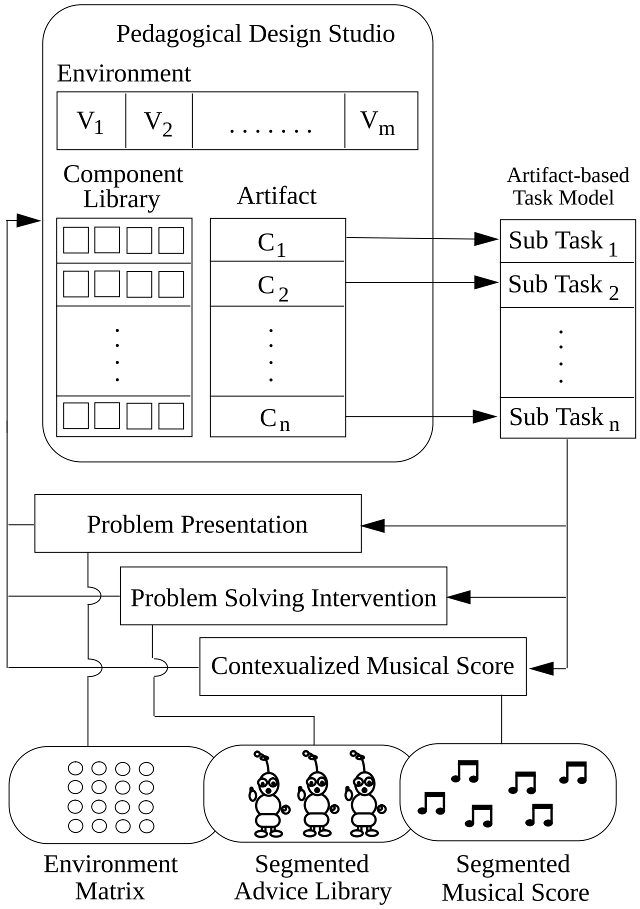

Figure 1: A Pedagogical Design Studio: Segmented task model and design studio

terface functionalities are tightly coupled to the task model, we discuss how they are exploited to produce customized learning experiences. The central concepts of pedagogical design studios are illustrated with DESIGN-A-PLANT, a learning-by-designing environment for botanical anatomy and physiology.[1] In DESIGN-A-PLANT, learners graphically assemble customized plants that will thrive in specified environments. We conclude by describing the lessons learned from an observational study of middle school students using DESIGN-A-PLANT.

INTERFACES FOR LEARNING-BY-DESIGNING

Learning-by-designing revolves around a carefully orchestrated series of design episodes in which the learner is given (1) an *environment* that consists of a set of environmental factors, and (2) an *artifact component library* containing the "building blocks" from which artifacts are assembled. The learner's task is to create an *artifact*, which is a compound object composed of components from the library, that can function successfully in the given environment. Regardless of the domain—whether the artifact being designed is a molecule, a house, or as we will see, a plant—learners are actively engaged in a process that requires them to grapple with fundamental issues in their given domain. Our hypothesis is that they will emerge from their experience with a deep appreciation for the rich, conceptual interconnections that define their subject matter.

In addition to the classic functionalities offered by microworlds [23, 12, 6, 13], learning-by-designing interfaces call for a look-and-feel that promotes constructivist learning. While each of the properties discussed below is important, our experience has shown that they are almost essential for learning environments which are to be used by children:

- *Direct Artifact Manipulation*: To actively engage the learner in the design process, the interface should offer a look-and-feel that supports direct manipulation of the artifacts. Rather than manipulating symbolic expressions or text, learners should graphically assemble the components into the complete artifacts. For example, to learn about botanical anatomy and physiology, students could interact with a direct manipulation interface to graphically assemble a plant from plant components (e.g., roots, leaves, and stems). The resulting experience will thus have an immediacy that would otherwise be unavailable.
- *Unified Problem Presentation / Problem Solving / Advice*: The interface should provide a single, unified visualiza-

[1] DESIGN-A-PLANT is a multi-disciplinary project involving computer scientists, multimedia designers, animators, and cognitive scientists. All of the 3D graphics and animations were designed, modeled, and rendered on Macintoshes and SGIs by a twelve-person graphic design team. DESIGN-A-PLANT runs on a Power Macintosh 9500/132.

tion of both problem presentation, problem solving, and advice. To enable learners to focus their attention on the design process—rather on the problem of learning to use multiple interfaces or interaction modes—a single visual presentation of problems, design workbench, and advice is critical. Switching between one interface for the presentation of the environmental variables, a second for the component library, a third for artifact assembly, and a fourth for advice would interfere significantly with problem solving. For young learners, it might halt the design process altogether.

- *Problem-Solving Flexibility*: The interface should enable learners to focus on any aspect of the problem-solving experience they wish. For example, in designing plants, they should be able to address the sub-problems of root, stem, and leaf selection in any order they wish. This flexibility gives rise to two types of complexity: on the learner's side, it increases the difficulty of the problem (though the increased freedom provides learning opportunities that take advantage of the design process); on the system's side, increased learner flexibility complicates diagnosis.
- *Customized Problem Presentation*: The interface should present environments that are most appropriate to the learner's design history. In particular, it should emphasize artifacts and environmental variables that (a) exercise concepts with which the learner has experienced difficulty, and (b) exhibit a degree of complexity that is not too great for the learner but is sufficiently complex to be challenging. For example, if a learner interacting with DESIGN-A-PLANT has exhibited a history indicating he or she has misconceptions about the structure and function of leaves in low sunlight environments, the system should endeavor to eliminate these misconceptions by presenting problems requiring the user to exercise his or her knowledge of these constraints. However, the number of active constraints in a particular design problem must not greatly exceed the learner's current abilities.
- *Goal-specific Intervention*: The interface should intervene with problem-solving advice when the learner experiences difficulty. Critically, the frequency and content of intervention should be appropriate for the particular aspects of the design task on which the learner is focusing, and advice should be relevant to the design goals currently being pursued. For example, in the DESIGN-A-PLANT system, if the learner is experiencing difficulties with particular aspects of root physiology, the system's remediation should focus on those particular problems.
- *Non-Invasive Diagnosis*: Both customized problem presentation and goal-specific intervention require that the system infer the learner's intentions. However, since continually interrupting learners to determine their current intent and to ferret out their misconceptions would interrupt the design process, diagnosis should be conducted non-invasively. For example, it would be against the spirit of a constructivist learning environment to prevent the learner from pursuing his or her design activities in order to issue a number of probes to detect precisely which misconceptions were active at a given time. In short, the interface should be crafted in such a way that, in the normal course of the learners design activities, the system can intuit his or her goals.

In addition to these characteristics, an attractive feature of a learning-by-designing interface is musical contextualization. While this property is by no means essential, by tailoring the voicing and melody of the soundtrack to the evolving design context, learning environments can subtlely reinforce problem-solving progress through the auditory as well as the visual channel.

PEDAGOGICAL DESIGN STUDIOS

Developing learning environments that provide all of the functionalities noted above poses a significant challenge. To address this problem, we have developed the *pedagogical design studio* framework, a unified approach to interactive learning systems that support learning-by-designing.[2] The key insight of this framework is two-fold: first, by carefully framing a *segmented* design problem, learning-by-designing interfaces can be crafted in such in way that learners' actions signal their intent and misunderstandings; second, a task model that is segmented in precisely the same manner as the design problem can be exploited to customize learning experiences. Developing a pedagogical design studio for a particular domain entails the following activities:

1. Define a *segmented design task* for the domain.
2. Define an *artifact-based task model* for the task whose segmentation mirrors the design task.
3. Develop a *design studio* interface that is directly mapped to the segmented task model.

We describe each of these activities in detail and illustrate their application with our experience in developing the DESIGN-A-PLANT learning environment. This is followed by a description of how design studios can exploit the task model to provide customized learning experiences.

Segmented Design Tasks

The much-studied task of design [16] involves the synthesis or combination of objects subject to constraints. Design tasks hold much appeal for learning because they are characterized by large problem spaces, but the complexity arising from the enormity of the problem spaces is difficult to manage, both from the learner's perspective, as well as from the perspective of developing a learning environment that is to perform diagnosis non-invasively. To combat this problem, the first step in developing a pedagogical design studio for a given domain is to define a segmented design task. Taking advantage of the "nearly decomposable" property of problems [20], we say that a design task is *segmented* if the following condition holds:

> **Design Segmentation**: The components of all artifacts are partitioned into distinct cells $P_1 \ldots P_n$, and any legitimate design will have exactly one component from each P_i.

[2] The long-term goal of this work is to develop a comprehensive understanding of intelligent learning environments that provide rich constructivist experiences. We are particularly interested in the potential educational impact of this technology on K-12. For the past two years, we have directed our studies at the middle school level.

To illustrate, consider a design problem in the domain of botanical anatomy and physiology.[3] We can define a segmented design task for botanical anatomy and physiology that has three partitions: roots, stems, and leaves. To design a plant, students much choose a particular kind of root system, a particular kind of stem system, and the type of leaves that their plant will have. Students are given an environment that specifies biologically critical factors in terms of qualitative variables. Environmental specifications for these episodes include the average incidence of sunlight, the amount of nutrients in the soil, and the height of the water table. Students consider these conditions as they inspect components from a library of plant components, each of which is defined by its physical characteristics such as length and thickness. Employing these components as their building blocks, students design a customized plant that will flourish in the current environment. Each iteration of the design process consists of inspecting the library, assembling a complete plant, and testing the plant to see how it fares in the given environment. If the plant fails to survive, students modify their plant's components to improve its suitability, and the process continues until they have developed a robust plant that prospers in the environment.

Segmented design tasks are a special (and manageable) case of the general design problem because the structural combinatorics are reduced. In contrast to the general case, the connections between candidate component types are fixed in advance. Segmented design tasks are more analogous to scheduling tasks than planning tasks. Because the type and structure of components are fixed in advance, the learner can focus his or her activities on selecting appropriate components for each partition. While only some design tasks are amenable to segmentation, for those that can be segmented, this approach can produce learning environments that lend themselves well to non-invasive diagnosis as shown below.

Artifact-Based Task Models

Dynamically customizing problem presentation and providing goal-specific interventions requires the system to recognize learners' intent, but plan recognition is a notoriously difficult problem [5, 7, 11]. To address it, we can exploit the segmentation of the design task to define an artifact-based task model whose segmentation mirrors that of the design task.

In contrast to more complex approaches to task modeling such as task-action grammars (TAGS) [18], production systems [17], or the myriad plan-based techniques [1], artifact-based task modeling reduces diagnostic complexity by exploiting (1) the segmentation inherent in the task itself, and (2) a design studio interface whose look-and-feel precisely mirrors the task segmentation. An artifact-based task model represents sub-tasks $S_1 \ldots S_n$, where each S_i represents the sub-task of making a decision about components of a particular type, namely, the components belonging to P_i:

- Each S_i records a history of design decisions made by the learner for that aspect of the design.

[3] Learning about this domain is known to be a very difficult task; investigating how children learn about photosynthesis, for example, has been the subject of much study [2, 3, 4].

Sub-Task	*Design History*	*Current Sub-task?*
Leaf Sub-task		
Stem Sub-task	Large, Thick, Branchy	√
	Small, Thick, Branchy	
	Small, Thin, Branchy	
	Small, Thin, Non-Branchy	
Root Sub-task	Deep, Thick	
	Deep, Thin	
	Shallow, Thick	

Table 1: Instance of the Artifact-Based Task Model for DESIGN-A-PLANT

- Each completed S_i records the most recent design decision with the selected component (from P_i).
- Some sub-task S_f, which the system believes the learner is currently focused on, is marked.

To illustrate, suppose a learner interacting with DESIGN-A-PLANT has begun to address the problems of what types of roots and stems to incorporate in her design for a particular environment that the system has presented to her. Furthermore, suppose she is currently considering issues bearing on stems, but has not yet begun to make decisions about leaves. The task model should be configured as shown in Table 1, with the design history for each sub-task recorded in temporal order (most recent first). The task model, together with the values of the environmental variables $E_1 \ldots E_m$ for a particular environment, represent the design problem and the learner's attempt to solve it.

Design Studio Look-and-Feel

After the task model has been defined, the next activity in creating a pedagogical design studio is to develop the design workbench interface. The workbench must allow users to attack the design problems flexibly: they must have the freedom to begin working on a sub-task, effortlessly move to new sub-tasks, revise design decisions in light of advice, and return to previously considered sub-tasks with ease. To simultaneously achieve this flexibility and to enable the system to perform diagnosis non-invasively, the interface state and its functionalities should be tightly coupled to the task model. This is accomplished by designing the interface so that it is visually and functionally segmented in the same manner as the task definition (and hence the task model). Operationally, it requires providing two functionalities, a *partition specifier* and a *component selector*. Learners use the partition specifier to indicate which sub-task they would like to address next and the component selector to communicate their decisions about particular components. To enable the system to properly maintain the task model, the interface must ensure that learners invoke the component selector for a particular partition only after they have first indicated the component type with the partition specifier. In addition to these key functionalities and their order of operation, the interface should provide a unified visual presentation of the problem, the design workbench, and the advice.

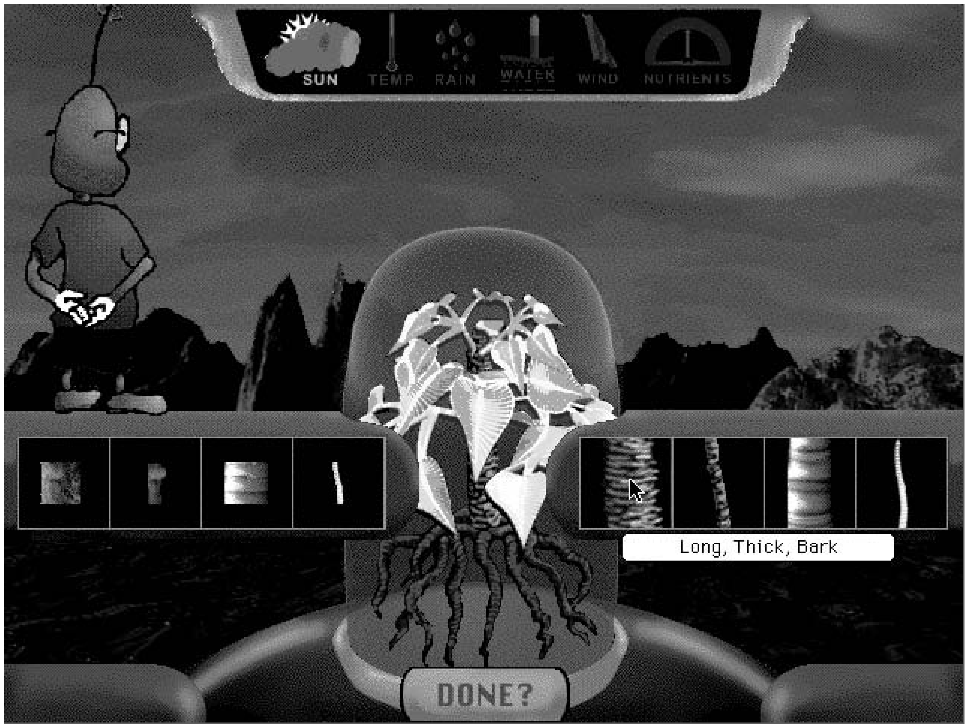

Figure 2: The DESIGN-A-PLANT Design Studio

To illustrate, the DESIGN-A-PLANT interface (Figure 2) was developed according to these criteria. The plant's target environment is presented with four types of redundant cues. First, an animated pedagogical agent in the form of a bug verbally describes the environments. Second, the "plant bubble" is located in a graphic of the landscape depicting the current environment. Third, each environmental factor is depicted iconically at the top of the screen. Fourth, learners may move the cursor to one of the icons representing an environmental factor, which gives a rollover textual description.

Artifact construction occurs in the "plant bubble" shown in the center of the screen. Learners graphically assemble plants by first positioning the partition specifier (the plant selection bar) vertically on the screen. This requires only a single mouse click. When the partition specifier is in the bottom-most position, the root library is display; when it is mid-level, stems are displayed; and when it is at the top, leaves are displayed. Learners then indicate design decisions by choosing a component of the selected type. This also is accomplished by a single mouse click. They may begin to address one sub-task, revise their decisions, move to other sub-tasks, or return to previous ones. Because all workbench actions are directly mapped to their corresponding sub-tasks, learners (perhaps unknowingly) signal their intent and progress through the natural course of the design process. When they believe their design is complete and correct, they click on the "Done" button at the bottom of the screen, and the system evaluates the learner's plant with respect to the given environment by searching for violations of constraints in the underlying domain model.

EXPLOITING ARTIFACT-BASED TASK MODELS FOR CUSTOMIZED CONSTRUCTIVE LEARNING

The benefits of maintaining the task model are realized in design experiences that are uniquely tailored for individual learners. In particular, the task model can be exploited to customize problem presentation, the soundtrack that accompanies problem-solving, and the advice that is presented to learners as they make and revise design decisions. We discuss each of these in turn, and illustrate their application in DESIGN-A-PLANT.

Customized Problem Presentation

To emphasize artifacts and environmental variables that exercise concepts with which the learner has experienced difficulty, the design studio can dynamically select environments. By inspecting the task model's design histories, the studio can not only present problems that have proved difficult for the learner in the past, but it can also control the level of complexity (as measured by the number of constraints) that the environment will exhibit. To do so, it exploits an *environment matrix* (Figure 3), where each element is an environment, which is a set of values for the environmental factors. Each column represents a particular environmental "intent," i.e., a particular sub-task to be exercised, and moving down the rows represents adding complexity. The first environment that all learners experience is the upper-left cell in the first row and first column. Progress through the matrix is always either straight across, which represents a new lesson at the same difficulty level, or diagonally down and across, which represents a new lesson at an increased difficulty level. If the learner reaches the far right column in the matrix, then they will return to the left-most column at a new difficulty level. When the learner reaches the bottom-most row, they

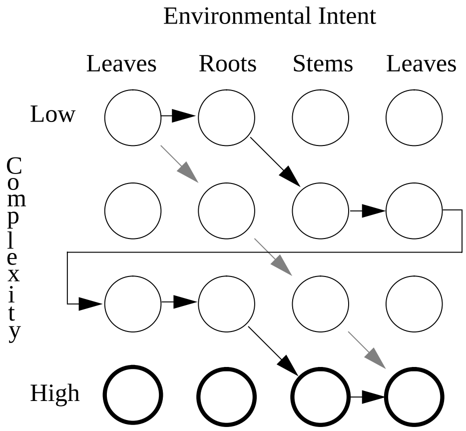

Figure 3: The DESIGN-A-PLANT Environment Matrix

are finished.

Determining which environment to present next is accomplished by inspecting the final state of the task model in the current environment. If the performance of the learner is sufficiently poor, as indicated by a series of incorrectly completed sub-tasks, then he or she will move horizontally across in the matrix. Otherwise, the move will be made diagonally. In Figure 3, two possible paths through the matrix are presented. The optimal path follows the dashed arrows which move diagonally from the starting environment to the most difficult environment. The path shown by the solid black arrows represents a more typical route that learners take. By monitoring the task model at each juncture, the studio is able to select environments that produce challenging and topical design experiences.

Contextualized Soundtrack Generation

The musical score should be dynamically contextualized to reflect the current state of the design problem. In addition, it should contain enough variety to keep any necessary repetition from becoming noticeable to the learner within the scope of a design episode, and it should be subtle enough to avoid distracting the learner from the design process. To contextualize the score, the design studio tracks the state of the task model and sequences the elements of the score so that as progress is made toward successful completion of sub-tasks, the number of voices added to the score increases.

DESIGN-A-PLANT uses this technique as follows. For variety, each environment has an independently constructed segmented musical score whose tempo and mood reflect the qualities of that environment. For example, the Alpine meadow soundtrack is based on a light quick melody played on panpipes reminiscent of Irish folk music, as opposed to the Southern marsh whose soundtrack is much more somber and moody. Within an individual environment, the score is divided into two primary melodies. One melody is played when the learner is actively engaged in design. The second melody is used to accompany animated advice. The two melodies, while different, share the same tempo, instrumentation and mood. The advice accompaniment can be a single instrumental song whose duration is as long as the longest possible animated advice sequence. Whenever the animated advice ends, the accompaniment song slowly fades. The more complex melody which accompanies active design is sequenced on the fly so that the music always mirrors the state of the design episode.

The segmented musical score contains each voice (instrument) which is part of the primary melody recorded in a separate track. Furthermore, four different variations of the melody are recorded: introductory, progressive, melancholy, and triumphant. Tempo is fixed within each segment, but can vary slightly between variations to reflect the desired quality. Volume is controlled separately. The introductory variation is played at the beginning of a design episode. As progress is made towards designing a successful artifact, the progressive variation is introduced. Successive failures result in the melancholy variation being played. Within each variation, the number of voices added to the score reflects the number of correct components in the partially completed artifact. Upon successful completion of a design episode, the triumphant variation is played. The music sequencer switches between variations on the melody and adds and removes voices so that the overall effect is one of a single continuous soundtrack whose melody, tempo, voicing, and volume reflect the current state of the design episode.

Delivering Customized Advice

To intervene with advice that is appropriate for the design goal that the learner is currently attempting to achieve, the design studio tracks all aspects of the task model. It uses the current sub-task indicator as a guide to the learner's current intent, and it uses the design history to provide advice about the effects of environmental variables on the design decisions that the learner is currently considering. Just like the soundtrack library, the library of multimedia advice is segmented by sub-task. The design studio uses the sub-task indicator to index into the collection of animated and audio advice clips, each of which is assigned to one or more sub-task indices.

DESIGN-A-PLANT's advice library is segmented into animated and audio clips that address particular types of design problems that learners experience with roots, stems, and leaves. Incorrectly completed sub-tasks suggest that the learner harbors particular misconceptions about the effect of environmental factors on their plant's components. These situations, as well as sub-tasks that are not completed after an extended period of time, trigger the design studio's advice system (Figure 4). Its library contains 30 animations and 160 audio clips containing advice that is presented by "Herman the Bug," an animated pedagogical agent [22]. The advice is selected to focus learners' problem solving [15], and the agent's behavior is sequenced in realtime to produce an intriguing lifelike effect [14]. Because the segmentation of the advice library mirrors the segmentation of the task model, interventions are appropriate for the learner's current focus of attention.

DISCUSSION AND CONCLUSIONS

Intelligent learning environments that support constructivism should provide active learning experiences that are customized for individual learners, and one particularly promising approach to constructivist learning is learning-by-designing, a paradigm we have been investigating in an ongoing project begun in 1994. Although customization of learning experi-

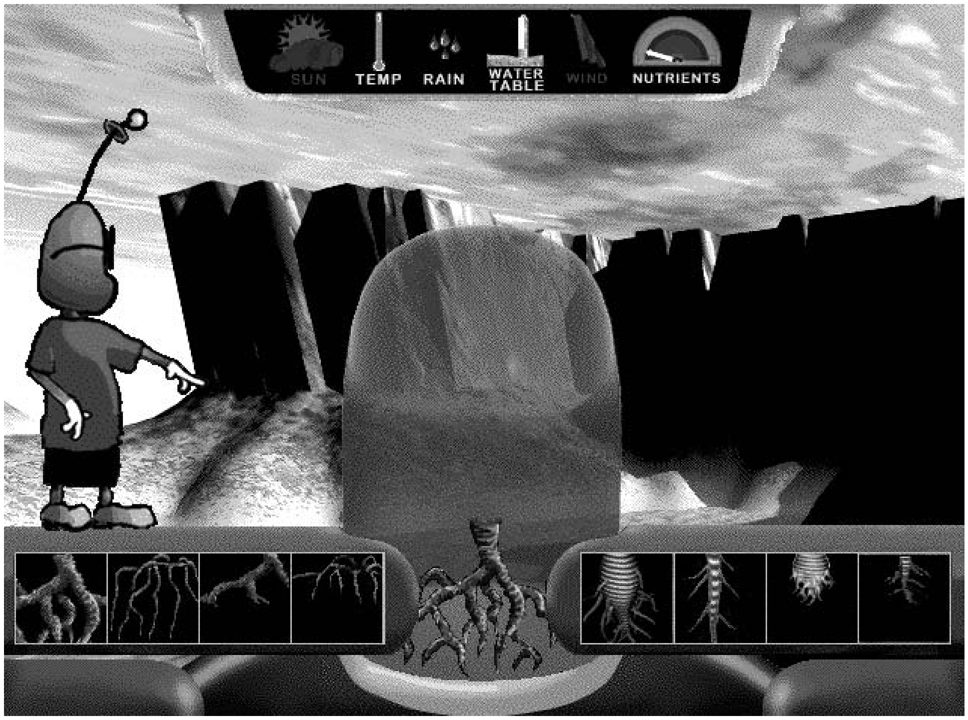

Figure 4: DESIGN-A-PLANT: Goal-specific Intervention

ences can take many forms, they all rely on an up-to-date representation of learners' intent. We have argued that by defining a segmented design task, it is possible to exploit an artifact-based task model to customize learning experiences, provided that the task model is tightly coupled to the functionalities of the design studio interface.

To explore these issues, we have instantiated the pedagogical design studio in DESIGN-A-PLANT, a learning-by-designing system for middle schoolers. DESIGN-A-PLANT has been the subject of two formative evaluations, one with thirteen middle school students from Martin Middle School of Raleigh, North Carolina, and one with ten middle school students from the Women in Science Mentoring Chapter of Raleigh, North Carolina. Each student interacted with the system for forty-five minutes to one hour.

While they are informal and observational, these studies suggest that learning-by-designing environments offer much potential for meaningful educational experiences that are individualized to the learners. The system as a whole was unanimously well received. The flexibility that the interface offers in moving between sub-tasks enabled students to approach design problems according to their own insights and experience. Perhaps equally importantly, the combination of the different forms of customization (customized problem presentation, contextualizing the musical score, and tailoring the problem-solving advice to the current sub-tasks) worked together synergistically to create a productive learning experience. The studies indicate that artifact-based task modeling constitutes an effective approach to non-invasive diagnosis in domains that are amenable to segmentation. The studies also echo the twin classic lessons from AI and HCI that the appropriateness of the representation and the interface in large part determine the performance of the system.

The encouraging results of the studies call for additional exploration of artifact-based task models. In particular, the modeling techniques need to be extended to cover two important problems in scaling up to more complex design tasks. First, designing more complex artifacts requires dealing with hierarchically decomposed components that have multi-level *partonomies*. Extending the modeling to cover deep partonomies has important implications for both the structure of the model itself, as well as for the interface. Second, designing more complex artifacts may necessitate the addition of temporal constraints on the performance of sub-tasks (and sub-sub-tasks, ...) Because pure segmentation, i.e., complete non-interaction, may be difficult to achieve in designing artifacts with complex partonomies, imposing temporal constraints on sub-tasks may significantly facilitate learning in these domains.

In summary, learning-by-designing is a promising technique that can be applied to a broad range of domains. In addition to domains such as architecture and engineering that have traditionally been taught with design methods, learning-by-designing can be applied to domains as diverse as biology (e.g., designing plants), chemistry (e.g., compound synthesis), or the social sciences (e.g., the popular Maxis SIM series). Given the encouraging results to date, we are particularly interested in quantitative measures of learning effectiveness and usability. To this end, we are currently embarking on a large-scale formal study to gauge precisely the cognitive effects of these learning environments.

ACKNOWLEDGEMENTS

Thanks to: the animation team of the North Carolina State University School of Design; the students in the Intelligent Multimedia Communication, Multimedia Interface Design, and Knowledge-Based Multimedia Learning Environments seminars; Chris Tomasson and the students in her seventh grade class at Martin Middle School of Raleigh, North Carolina; the Raleigh chapter of the Women in Science Mentoring Program; and Charles Callaway for helpful comments on a previous draft of this paper. Support for this work was provided by the IntelliMedia Initiative of North Carolina State University and donations from Apple and IBM.

REFERENCES

1. J. Allen, J. Hendler, and A. Tate, editors. *Readings in Planning*. Morgan Kaufmann, San Mateo, CA, 1990.
2. R. Amir and P. Tamir. In-depth analysis of misconceptions as a basis for developing research-based remedial instruction: The case of photosynthesis. *The American Biology Teacher*, 56(2):94–100, 1994.
3. M. Barker and M. Carr. Teaching and learning about photosynthesis, Part 1: An assessment in terms of students' prior knowledge. *International Journal of Science Education*, 11(1):49–56, 1989.
4. M. Barker and M. Carr. Teaching and learning about photosynthesis, Part 2: A generative learning strategy. *International Journal of Science Education*, 11(2):141–152, 1989.
5. S. Carberry. Plan recognition and its use in understanding dialog. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 133–162. Springer-Verlag, Berlin, 1989.
6. E. Cauzinille-Marmeche and J. Mathieu. Experimental data for the design of microworld-based system for algebra. In H. Mandl and A. Lesgold, editors, *Learning Issues for Intelligent Tutoring Systems*, pages 278–286. Springer-Verlag, New York, 1988.
7. J. Chu-Carroll and S. Carberry. A plan-based model for response generation in collaborative task-oriented dialogues. In *AAAI-94: Proceedings of the Twelfth National Conference on Artificial Intelligence*, volume 1, pages 799–805, 1994.
8. D. C. Edelson. Learning from stories: Indexing and reminding in a socratic case-based teaching system for elementary school biology. Technical Report 43, The Institute for the Learning Sciences, Northwestern Univeristy, Evanston, Illinois, July 1993.
9. M. Eisenberg and G. Fischer. Programmable design environments: Integrating end-user programming with domain-oriented assistance. In *CHI '94: Human Factors in Computing Systems: Celebrating Interdependence*, pages 431–437, 1994.
10. M. D. Gross. Design and use of a constraint-based laboratory for learning design. In R. W. Lawler and M. Yazdani, editors, *Artificial Intelligence and Education*, volume 1, pages 167–181. Ablex, Norwood, NJ, 1987.
11. R. W. Hill and W. L. Johnson. Situated plan attribution. *Journal of Artificial Intelligence in Education*, 6(1):35–66, 1995.
12. J. D. Hollan, E. L. Hutchins, and L. M. Weitzman. STEAMER: An interactive, inspectable, simulation-based training system. In G. Kearsley, editor, *Artificial Intelligence and Instruction: Applications and Methods*, pages 113–134. Addison-Wesley, Reading, MA, 1987.
13. R. W. Lawler and G. P. Lawler. Computer microworlds and reading: An analysis for their systematic application. In R. W. Lawler and M. Yazdani, editors, *Artificial Intelligence and Education*, volume 1, pages 95–115. Ablex, Norwood, NJ, 1987.
14. J. C. Lester and B. A. Stone. Increasing believability in animated pedagogical agents. To appear in *Proceedings of the First International Conference on Autonomous Agents*.
15. J. C. Lester, B. A. Stone, M. A. O'Leary, and R. B. Stevenson. Focusing problem solving in design-centered learning environments. In *Proceedings of the Third International Conference on Intelligent Tutoring Systems*, pages 475–483, 1996.
16. D. Navinchandra. *Exploration and Innovation in Design: Towards a Computational Model*. Symbolic Computation. Springer-Verlag, New York, 1991.
17. A. Newell. *Unified Theories of Cognition*. Harvard University Press, 1990.
18. S. Payne and T. Green. Task-action grammars—a model of the mental representation of task languages. *Human-Computer Interaction*, 2:99–133, 1986.
19. J. Piaget. *The Construction of Reality in the Child*. Basic Books, New York, 1954.
20. H. A. Simon. *The Sciences of the Artificial*. MIT Press, Cambridge, MA, 1981.
21. M. Smith. CONNIE: An intelligent learning environment for creative tasks based on negotiation of constraints. In *Proceedings of the Artificial Intelligence in Education Conference*, pages 397–404, 1995.
22. B. A. Stone and J. C. Lester. Dynamically sequencing an animated pedagogical agent. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 424–431, 1996.
23. P. W. Thompson. Mathematical microworlds and intelligent computer-assisted instruction. In G. Kearsley, editor, *Artificial Intelligence and Instruction: Applications and Methods*, pages 83–109. Addison-Wesley, Reading, MA, 1987.