Helping Instructors Scaffold Students' Design of Educational Technology Projects

Chris DiGiano
Michael Chorost
Mark Chung
Jeff Huang

SRI International
333 Ravenswood Avenue
Menlo Park, CA, USA
[chris.digiano, mark.chung, jeff.huang]@sri.com, michael@michaelchorost.com

Abstract: The design of effective learning technologies is a complex and ill-structured practice that can be difficult to teach to future generations of designers. Here we discuss our development of a tool that scaffolds students' design activity by prompting them to consider, in writing, many of the key issues that professional designers confront when planning a new project. This tool, called Gorp ("Gallery, Organizer, and Repository of Projects"), is a Web-based application used by universities across the country. Gorp has evolved to scaffold critical and often overlooked aspects of learning technology design through its organization of descriptive categories. The system is also designed to encourage students to collect feedback on their projects and iterate them through a process of testing and revision.

Introduction

University instructors who teach courses in learning technology design face a number of important problems that have not been well-addressed in the research to date. As Craig et. al. (2000) note, design is a complex and ill-structured domain, with few obvious and easy answers. Since the same is true for education, the problem is compounded when the topic is learning technology design. Students need considerable guidance in identifying user needs, specifying proposed designs in detail, and iterating upon them through a process of feedback and revision. It is critically necessary to provide instructors with tools that they can use to scaffold students' work over a semester or quarter. In this paper, we describe an online tool intended to assist in such scaffolding. The tool – called the "Gallery, Organizer, and Repository of Projects", or "Gorp" for short, was designed within the context of an NSF-funded project called Training and Resources for Assembling Interactive System, or TRAILS. (The name "Gorp" was chosen to fit in with the TRAILS metaphor, since "gorp" also means a hiking snack, that is, "Good Old Raisins and Peanuts.") See http://trails-project.org/webapps/gorp/Gorp.do to view the GORP library.[1]

Gorp was developed under the aegis of TRAILS, a multiyear, NSF-sponsored effort to broaden and support the pool of talent available to create technology for K-12 education. To TRAILS, examples of educational technology include simulations, adaptive tutorials, interactive exhibits, and educational games. Our approach focuses on engaging teams of university students in creating learning technologies through project-based design courses. TRAILS currently sponsors and supports four pilot courses across the U.S., which we hope will provide models for future learning technology design courses. Through these courses TRAILS intends to have three major effects: to better prepare tomorrow's designers of learning tools, to

[1] For additional papers resulting from TRAILS courses and research, look up the other papers in this ED-MEDIA symposium: Mercier, Booker, & Goldman, "Bringing collaboration front and center in a cross-disciplinary design course"; Repenning & Clayton, "Playing a game: The ecology of designing, building and testing games as educational activities"; and Hoadley & Cox, "Educating Reflective Learner Centered Designers."

better prepare the classroom teachers who will deploy such tools, and—by publishing select course projects— to generate new tools for K-12 education. TRAILS is a follow-on to the ESCOT project, which identified key advantages of teaming developers and educators in design learning technologies (see Roschelle & DiGiano, 2004).

Gorp is just one aspect of TRAILS, although it has become a central one. It is now used by most TRAILS-supported courses as a way of guiding students through the process of conceptualizing and developing software over a semester's time. It does this by requiring them to address three critical needs in training students to do learning technology design: (1) reflecting on learner needs, (2) design decomposition, and (3) iteration.

On the surface, Gorp is simply a Web application for TRAILS students to document their projects, mainly through textual descriptions. A project team can create a Gorp entry by following a simple wizard-style interface to populate a set of standard fields that describe their work, and revise their descriptions over time as their program is developed and tested. Students do so by entering and saving detailed descriptions of various aspects of the program, prompted by category headings and subheadings. The categories given ("Teaching and Learning," "Technology," and so forth) are intended to take novice designers through a process of addressing the issues that experienced educational designers usually deal with when undertaking new projects. Moreover, it enables them to refine their answers over time over a multistage design process (to be described in a following section.)

We have carefully organized the fields and other mechanisms in the system to encourage reflection, design decomposition, and development iteration. Thus Gorp is in line with calls for designers to think about specific design factors when they are designing a tool; for example, Alessi and Trollip (2000) suggest that designers of simulations consider 13 specific factors, such as "delivery mode," "instructional strategy," and "motivators."

In actual use, the nature of student projects vary in approach and granularity across courses. Some student teams limit themselves to teaching a particular math subtopic such as linear graphing, while others are more ambitious and attempt to cover whole subject areas such as reading Japanese characters. Most projects result in at least a prototype product, but in some cases students focus on detailed design documentation that could be used to implement a prototype later. So far, about half of the students in TRAILS courses have produced learning tools that draw from specific techniques used in computer gaming to engage student users.

The screenshot below shows a top-level view of Gorp, displaying several student teams' programs.

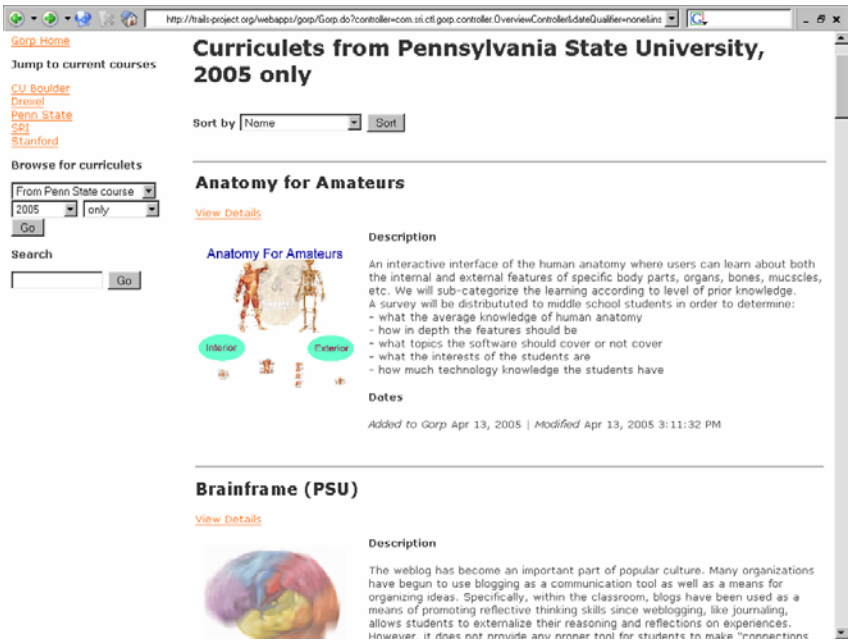

Figure 1: The overview screen of Gorp.

When a user clicks on "View Details," they see a detailed description of a program, organized by headings and subheadings (see below.) These can be and often are changed by the instructor, as we will describe below.

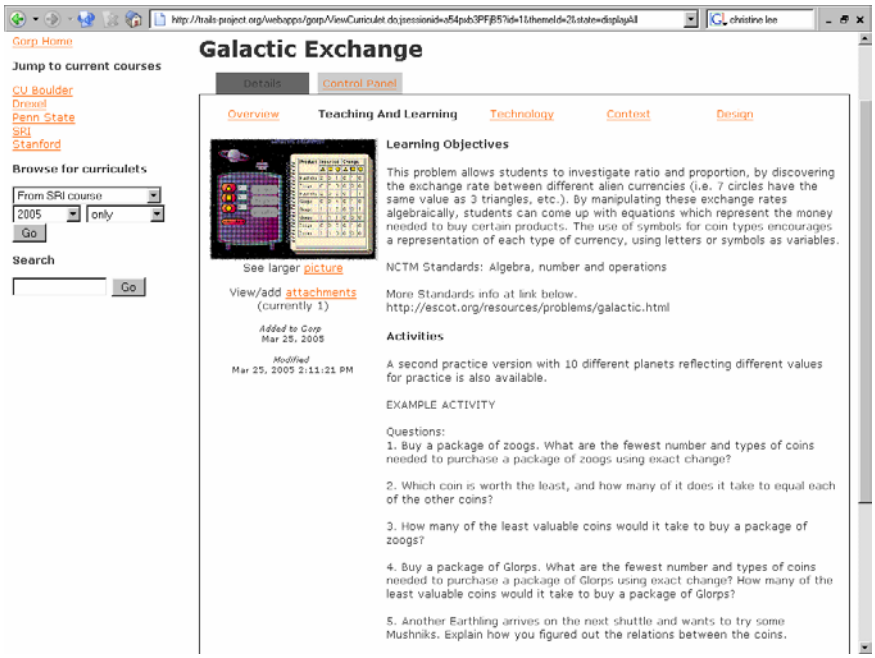

Figure 2: Viewing one entry in detail.

Gorp is therefore a both repository and a structured exploration of various aspects of learning technology design. It has two other goals as well. We designed Gorp to be an illustration of past students' projects and exemplars by professionals that would motivate new students to do high quality work and get an early sense of the expected scope of projects. This goal is in alignment with an instructional "bundle" in Fincher, Petre and Clark (2001) aptly titled "Well, *They* Managed." This bundle recommends that instructors "use deliverables from previous years to help students get an idea of what they have to do" (p. 258). The availability of a wide range of examples also enables students to engage in contrasting-cases exercises, e.g. comparing programs to each other, if the instructor wishes (Bransford et. al., 2000).

We had originally considered several off-the-shelf software packages for supporting collaborative work between students before deciding to develop Gorp, a custom-made solution, from scratch. We considered several groupware applications and course management systems, but they seemed heavyweight, difficult to use and customize, and likely to meet resistance from instructors already invested in competing systems. What we thought we wanted at that time was a tool that was simple to use but also likely to afford complex behaviors (Legos and HTML are good examples of the kind of simplicity/complexity effect we had in mind.) Therefore we considered using Wikis, which let users create simple Web pages very easily and link them to each other.[2] In discussing CoWeb, a software package similar in philosophy to Wikis,

[2] Specifically, we used Tiki (http://tikiwiki.org) as our Wiki solution because it had the most developer activity of competing open source Wiki implementations hosted on Sourceforge (others considered included PHPWiki, Swiki, Twiki, and VQWiki), and had a long list of features that we thought could potentially be useful to our TRAILS courses, such as discussion forums, file galleries, commenting, polling, access control, and a plugin architecture.

Guzdial et. al. (1999) observed that its "response from teachers has been to invent a wide range of activities, with the same simple tool" (p. 211). Craig et. al. (2000) suggest that CoWeb is useful for teaching in complex and unstructured domains such as design because it gives teachers and students a high level of control and flexibility in managing class activity. Another rationale that could be offered is that in any activity, the tool should mirror the domain: in this case, unstructured tools to allow for improvisation while working in unstructured domains.

However, Wiki-style solutions didn't meet our needs. They were *too* unstructured, whereas we found ourselves needing to provide concrete and specific guidance to teachers and students. Two early TRAILS courses tried a Wiki, but one instructor later moved to a course management system (Blackboard) because it was supported by her department, and the other instructor wanted more control over functionality and the ability to integrate different kinds of content (such as images and applets) onto a single page, so he created his own custom solution. We thus abandoned a Wiki-type solution and went in the other direction: instead of building a highly unstructured tool, we built a relatively structured one where the flexibility lay in being able to modify the structure of the activity rather than invent wholly new activities. If one thinks of interactive tools as falling in a spectrum from unstructured to structured, Wikis and CoWeb would fall at one end and Gorp at the other.[3] We knew that instructional designers explicitly articulate many things – learner needs, learning objectives, associated non-computer activities, technology requirements, alignment to educational standards, user testing strategy, and so on – before writing a single line of code, and that they frequently change their ideas during the process of development. We decided it would be of most value if we scaffolded that process of specification, reflection, and iteration in a relatively directive way. Other researchers have also built tools intended to direct and guide classroom work over time, such as Loh et. al. (1998), and we similarly wanted to enable students to capture their work in such a way that they could reflect upon it, iterate it, and display it for viewing by others both during the semester and afterward. Gorp's first version was released in Spring 2004.

Challenges in Designing Gorp

Gorp is central to the TRAILS project because it helps disseminate effective teaching practices about design by instantiating them in an easily accessible interactive medium. By making Gorp available to any teacher who wishes to use it, even for courses not directly supported by TRAILS, we hope to spread effective teaching practices about design in a way that is scalable and simple.

The goal of creating a tool that could be used in diverse teaching circumstances presented major challenges for the TRAILS team. Instructors often have very different ideas of what constitutes good software design, so creating a tool that imposed a single vision of effective design was unlikely to work. As Fincher, Petre, and Clark (2001) point out, practices developed by one institution are rarely if ever adopted verbatim by others. The particular circumstances of each instructor's environment require adaptation and customization. In fact, Fincher, Petre, and Clark suggest that transfer of practices can *only* be successful when the recipient customizes a practice to fit their needs (pp. 171-2). Therefore, we wanted to create a tool that offered a useful organizing structure yet also gave instructors the opportunity to modify that structure.

The need for this kind of flexibility quickly became clear when we tested early versions of the default Gorp schema with instructors at Stanford and the University of Colorado at Boulder. They responded that while they liked certain categories, they disagreed with others and wished to create new ones that we hadn't thought of. We modified Gorp so they could change the schema in any way they liked. This, we hoped, would offer instructors the right balance between convenience and flexibility. We also hoped that it would scaffold students' activity without either unduly constraining them nor overwhelming them with something that was too open-ended.

---

[3] To give an idea of what would be at the middle of that spectrum, consider SenseMaker, a program developed by Phillip Bell at UC-Berkeley (see Bell, 1997). SenseMaker scaffolds middle school students' engagement with scientific thinking by enabling them to group hypotheses and supporting evidence in "claim frames." Like CoWeb, it allows users to place entries where they wish in relationship to other entries, but like Gorp, it requires that those entries be about specific things and fit a specific format.

The Gorp Schema

Gorp's core is its schema – its design framework for capturing and thinking about the role of a learning technology and its features. It consists of a default set of boxed ideas, so to speak, that cover the key issues that educational designers usually consider when spec'ing out a product. Its major headings include the following:

- Overview
- Teaching and Learning
- Technology
- Context
- Design

Each of those headings has a number of subheadings. For example, the "Teaching and Learning" heading has subheadings such as "Learning Objectives," "Activities," "Assessment", and "Alignment to curricular standards." The "Technology" heading has subheadings including "Development Platform" and "Testing and Feedback." (The full Gorp schema is given in an appendix.) The meanings of each item are now amply illustrated with numerous examples of applications created by five semesters of TRAILS courses. Gorp therefore offers both an abstract schema and concrete examples of its application, facilitating conversation and reflection (Schon, 1992).

Note that the schema is multidisciplinary. To complete it, a team has to know about pedagogy, design, and technology, at the very least. In the real world, educational designers usually have to create not just software but also documentation, assessment materials, and alignments to state or national educational standards. Therefore, in TRAILS courses, student teams are often (and ideally) multidisciplinary, consisting of computer science students, education students, and graphic design students. Multidisciplinary teams pose challenges in the classroom just as in the professional world: team members are conversant in their own discipline but rarely those of the others'. Computer science students are rarely familiar with the real-world need of meeting government-mandated educational standards; conversely, education students are rarely familiar with the process of software design and testing. Hoadley and Kim (2003) have pointed out that universities have generally failed to graduate students who are cross-trained in software development and educational theory and practice. Gorp is designed, therefore, to use the language of several disciplines and to offer student teams the framework for a common language in which disparate team members can productively talk about learning in a technological context.

The Customizability of Gorp's Schema

As mentioned earlier, the schema is customizable. Teachers can rename, add, and delete major headings and subcategories. This was critical to its adoption. We wanted teachers to be able to modify the taxonomy to meet their own needs, or they would reject the tool as being too rigid, but we also wanted there to be enough standardization between classes that students' work would be at least approximately comparable.

This presented us with a problem. Did we want to "lock down" certain headings and subcategories to ensure at least minimal comparability between courses, or to give instructors full freedom of customization in order to enhance its odds of being adopted? We chose to do the latter. As can be seen in the screenshot below, one instructor made fairly substantial changes in both the headings and the subcategories.

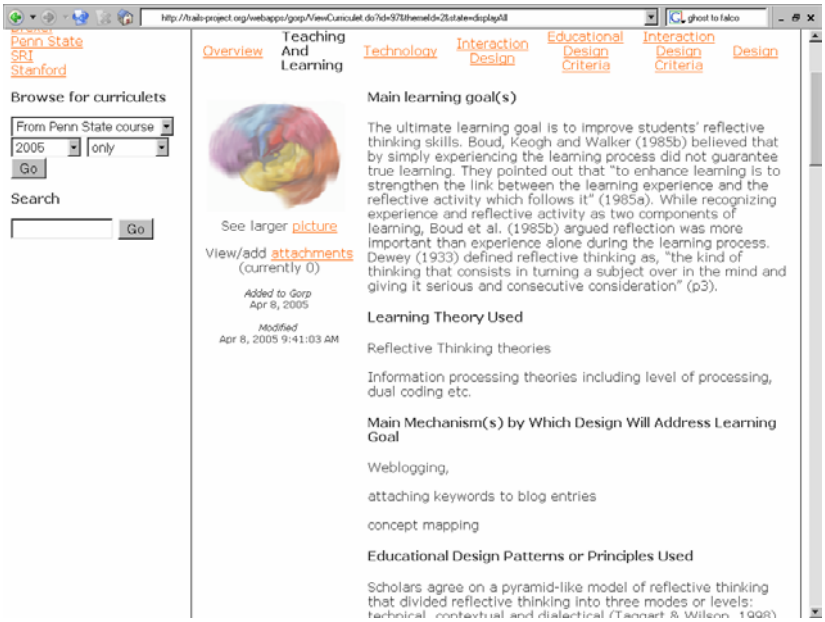

Figure 3: One instructor's changes to part of the Gorp schema.

However, it turned out that there was enough tacit agreement between instructors to ensure at least approximate consistency between courses. Every instructor retained the "Teaching and Learning" heading, for example, although one changed the name to "Curriculum" – a minor difference of terminology – and another created an entirely new heading titled "Games" with a number of subheadings. It appears that so far, the Gorp schema has functioned successfully as a core schema which instructors can customize without substantially undermining comparability between courses. We think that the default schema successfully packages the essentials, and that instructors are tweaking and amending it rather than fundamentally changing it.

Gorp's Evolution In Response to Instructor Feedback

Our ideas for how Gorp can and should be used in the classroom have changed, sometimes in surprising ways. We were open to modifying the code in response to teacher requests. One major modification had to do with representing student work over time.

In its early incarnations, Gorp had no time dimension; it assumed that students would keep overwriting their entries until arriving at final versions at the end of the semester. Our Drexel partners pointed out that Gorp would be much more useful to them if it supported a phased design process, where students could store their intermediate answers such that their evolution could be reviewed at the end of the semester. This made sense to us, because most TRAILS courses have their students undertake development work in phases, starting with a needs analysis and then moving to design concepts, paper prototypes, user testing, working prototypes, and so on. According to Loh et. al. (1998), the capturing of intermediate work products supports reflective inquiry by documenting its stages over time. Having access to intermediate work enables students and teachers to consolidate their learning from experience as they monitor and evaluate their progress.

Therefore, in November 2004, we undertook a major revision of Gorp to add a time dimension to the schema. Phases were visually represented by horizontal tabs at the top of the display area. When a team finished a phase, they could create another one and keep going. When that happened, the entries of the previous phase were locked down to prevent further changes and then copied to the new phase. An example of the revised design is shown below.

Figure 4: Phases in Gorp.

Other changes implemented at instructor request include adding the ability for outside experts to give written feedback on student entries. Drexel asked that outside users be disallowed from viewing its students' entries until the final stage was complete, and that only the final stage be visible at that point. The rationale was that draft work should be protected from scrutiny until it was considered ready for public viewing. Another instructor asked that a function be added which let instructors view all student entries in a single, printable page, to facilitate easy review and grading at the end of the semester.

In fact, the meaning of the software's name has itself changed in response to feedback from course instructors. Originally, "Gorp" stood for "Gallery Of Reusable Projects", as we had anticipated that students would mine previous work and extend it. However, instructors expressed little interest in having their students creating derivative projects, and regarded such projects as being difficult to evaluate. Students also placed considerable value on having ownership of their own design ideas. After it was used by two courses, therefore, the meaning of the acronym was changed to "Gallery, Organizer, and Repository of Projects."

Conclusions

In this paper we've outlined an approach to designing a tool that helps students design tools – a curiously self-reflexive undertaking. We had originally envisioned an almost completely unstructured tool, but experience taught us that a more directive approach that explicitly scaffolded students' engagement with the concrete issues of educational software design was likely to be of more value. The core of Gorp is its schema, which, as noted, was tweaked and extended by various instructors but not fundamentally changed. One of its most important aspects is its multidisciplinary focus, guiding teams in addressing concerns from computer science, education, and design in a unified way. The addition of support for a phased design process considerably extended Gorp's functionality in new directions. In future publications, we expect to offer a detailed evaluation of Gorp's efficacy at its stated goals.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0205625. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Alessi, S.M., & Trollip, S.R. (2000). *Multimedia for learning: Methods and development* (3rd ed). Boston, MA: Allyn & Bacon.

Bell, P. (1997). Using argument representations to make thinking visible for individuals and groups. In R. Hall, N. Miyake, & N. Enyedy (eds.), *Proceedings of CSCL '97: The Second International Conference on Computer Support for Collaborative Learning*, pp. 10-19. Toronto: University of Toronto Press. Retrieved February 7, 1998 from http://www.oise.utoronto.ca/cscl/papers/bell.pdf.

Bransford, J.D., Brown, A.L., & Cocking, R.R., Eds. (2000). *How people learn: Brain, mind, experience, and school*. Washington, D.C.: National Academy Press.

Craig, D., ul-Haq, S., Khan, S., Zimring, C., Kehoe, C., Rick, J., & Guzdial, M. Using an Unstructured Collaboration Tool to Support Peer Interaction in Large College Classes. In B. Fishman & S. O'Connor-Divelbiss (Eds.), Fourth International Conference of the Learning Sciences (pp. 178-184). Mahwah, N.J.: Erlbaum.

Fincher, S., Petre, M., & Clark, M. *Computer Science Project Work: Principles and Pragmatics*. London: Springer, 2001.

Guzdial, M., Realff, M., Ludovice, P., Morley, T., Kerce, C., Lyons, E., & Sukel, K. (1999). Using a CSCL-driven shift in agency to undertake educational reform. In C. Hoadley (ed.), *Proceedings of CSCL '99,* pp. 211-217. New York, NY: Association for Computing Machinery.

Hoadley, C., & Kim, D. E. (2003). Learning, design, and technology: Creation of a design studio for educational innovation. In A. Palma dos Reis & P. Isaís (Eds.), *Proceedings of the IADIS International Conference e-Society 2003* (pp. 510-519). Lisbon, Portugal: International Association for the Development of the Information Society (IADIS).

Hoadley, C. & Cox, C. Educating reflective learner centered designers. In *Proceedings of ED-MEDIA: World Conference on Educational Multimedia, Hypermedia & Telecommunications*. June 27-July 2, 2005, Montreal, Canada.

Loh, B., Radinsky, J., Russell, E., Gomez, L. M., Reiser, B. J., & Edelson, D. C. (1998). The Progress Portfolio: designing reflective tools for a classroom context. In *Proceedings of CHI 98.* pp. 627-634. Reading, MA: Addison-Wesley.

Mercier, E., Booker, A., & Goldman, S. (2005). Bringing collaboration front and center in a cross-disciplinary design course. In *Proceedings of ED-MEDIA: World Conference on Educational Multimedia, Hypermedia & Telecommunications*. June 27-July 2, 2005, Montreal, Canada.

Quintana, C., Reiser, B.J., Davis, E.A., Krajcik, J., Fretz, E., Duncan, R.G., Kyza, E., Edelson, D., & Soloway, E. A scaffolding design framework for software to support science inquiry. *Journal of the Learning Sciences* 13:3.

Repenning, A. & Clayton, L. Playing a game: The ecology of designing, building and testing games as educational activities. In *Proceedings of ED-MEDIA: World Conference on Educational Multimedia, Hypermedia & Telecommunications*. June 27-July 2, 2005, Montreal, Canada.

Roschelle, J., & DiGiano, C. (2004). ESCOT: Coordinating the influence of R&D and classroom practice to produce educational software from reusable components. *Journal of Interactive Learning Environments*, 12(1-2), pp. 73-107.

Underwood, J., Hoadley, C., Stohl, H., Hollebrands, K., DiGiano, C., Renninger, K.A. IDEA: Identifying design principles in educational applets. Educational Technology Research and Development (to appear).