# Toward a Technology Community in the Learning Sciences

*Chairs*
James Slotta, University of Toronto, Canada, jslotta@oise.utoronto.edu
Turadg Aleahmad, Carnegie Mellon University, Pittsburgh, PA, turadg@cs.cmu.edu

*Panelists*
Chris Quintana, Assistant Professor of Education, School of Education, University of Michigan
Jeremy Roschelle, Director, Center for Technology in Learning, SRI International
Turadg Aleahmad, Phd Student, Human Computer Interaction Institute, Carnegie Mellon University
Wouter van Joolingen, Associate Professor, University of Twente

*Discussants*
Nikol Rummel, Assistant Professor of Psychology, University of Freiburg
Chris Hoadley, Associate Professor of Education, New York University
Eleni Kyza, Lecturer, Dept. of Communication and Internet Studies, Cyprus University of Technology

**Abstract:** CSCL depends critically on technology development, yet the process of development is not much discussed in a research context. This panel is aimed at helping the CSCL community think more critically about how it develops technology and uses it.

## Introduction

CSCL research and implementation depend critically on technology development, yet the process of development is not much discussed in a research context. This panel is aimed at helping the CSCL community think more critically about how it develops technology and uses it. Researchers in the learning sciences must be able to use technology with agility in order to create materials, manage complex pedagogical structures, collect data, analyze complex patterns, and report their findings. They must be able to exchange these materials in order to promote replication and innovation in the field. Finally, they must connect their materials to the public discourse on standards, re-use, and open source licensing. It is to our advantage as a field if we move toward a more coherent set of practices and internal standards regarding our own uses of technology. Further, we must enable our young investigators to get involved in such research, reducing the absurd level of collaborative and development overhead that currently confronts them. The average level of human resources that are available to any investigator in the field are certainly less than one full-time programmer – most likely closer to a quarter-time programmer. How can we enable investigators to move forward confidently in their research with such limited staff for developing technology infrastructure for their research?

This panel is proposed to discuss various efforts that have been made and are currently underway to address this matter, and to engage the community in a discussion about the nature of the problem. We will look at this problem through the themes of re-use, adaptability, easy of use and interoperability. The panel and audience will debate the costs and benefits of designing towards these goals and specific potential remedies such creating and participating in an open-source community of education research technology development. Is it reasonable to aspire to the creation of a commons of education research software components that can be set up and configured by any technology specialist in a short amount of time? Can they be made simple enough to configure and to author new materials and activities that a graduate student with some technical orientation could learn to do so without distracting from their research questions?

As a field, we have several prominent examples of large-scale efforts to develop scalable, reusable, interoperable software, which have met with some limited success. There are many reasons why previous efforts haven't resulted in an easy-to-use framework for new investigators, and these will be discussed by the panel. Perhaps more interesting, there are new emerging technologies and philosophies that may help our efforts become more successful. The goal of this panel is to understand the problems confronting our community in relation to technology frameworks and to begin articulating some of the possible solutions or responses to those problems.

To that end, we also address the issue of technology developers as a valuable element of our intellectual community. As the world of technology evolves, it offers new insight into the powerful mechanisms of collaboration, aggregation of knowledge, community, social and semantic networking. Researchers are captivated by the new metaphors of Web 2.0, the promise of open source and open content, and new forms of human computer interaction made popular by the Wii, the iPhone and multi-touch surfaces. Yet it is our technology specialists who lead the way in developing such innovations. In the past ten years, our technology staff have shifted from being "programmers who implemented our designs" to being co-designers and even colleagues. This is an exciting development, pointing to new intellectual capacity in our field that must be supported. This panel will recognize the opportunities that could come from nurturing a community of technology developers, who might otherwise work in isolation – helping them connect with peers, exchange ideas, and co-develop common resources.

## Structure of the Panel

The panel will include two main phases: First, we invite representatives from four prominent technology development projects from the learning sciences to offer insight and reflections about their own trajectory in developing tools and materials that are aimed at supporting re-use and interoperability. Each panelist will be asked to speak for 10 minutes regarding several formative questions: (1) Describe your interests and achievements in developing a framework for re-use and interoperability. (2) Describe your success in supporting other researchers in the learning sciences to adopt this framework. (3) Describe the key challenges faced by your technology framework. (4) Describe how your technology developers have been involved in any sort of intellectual community. Next, we invite 3 individual researchers whose research requires technology-enhanced materials to discuss with the panel their general interests and specific challenges they've experienced, including their frustrations in using existing software, failing to find appropriate software or having to develop new software with limited technology development resources. These discussants will join the panel in a discussion to scope the problem and evaluate possible remedies. The following sections describe the four projects included in the panel and their perspectives on the four guiding questions.

### 1. Symphony and HI-CE - Chris Quintana & Elliot Soloway

We have developed large, integrated constructivist software environments for middle school and high school learners (e.g., Symphony, the Digital IdeaKeeper). Each of these software environments involved integrating a set of smaller software components within innovative, scaffolded interfaces that aim to support learners with different inquiry practices. In order to support such development, we have explored new component frameworks, such as our Symphony2 framework. Other similar frameworks have also been developed with the promise of providing a single technology infrastructure that developers and researchers could use to easily create new learner-centered software. However, while such component frameworks have been well intentioned, there are still significant challenges involving the component size, customization, and usability of such frameworks. We have seen the difficulty of using software components and frameworks to assemble different individual software parts into a larger software whole. These issues leave us questioning whether these software frameworks have truly provided our community with the leverage and support needed for research and development of learning technologies.

One issue involves the grain size of the components that can be integrated within a given framework. In Symphony2, we attempted to integrate larger software components (e.g. more general purpose modeling applications, visualization applications, etc.) These larger components provided more flexibility, but developing and integrating such large components was difficult. Other frameworks have components at a smaller grain size (e.g., very specific modules that have a single functional purpose). These components may be easier to develop and integrate, but may be too limited or specific in scope to assemble into a richer software project.

A second issue involves the customization capabilities of a given framework. Developers may find a component that is "close, but not quite" in terms of their requirements, but then find it difficult to get "under the hood" and revise the component so it fits their project needs. Similarly, there are customization issues involved in developing supportive interfaces for learning technology projects. It is still difficult for developers to add and customize scaffolding features along with functional components in the larger software project.

Finally, a third issue involves the usability of these frameworks—how easy is it for developers to put pieces together and create a larger piece of software. Developers have found it difficult to use software frameworks because of the different integration approaches and vocabularies that each uses. Additionally, there is the difficulty in learning and using the languages that underlie these frameworks. For example, some frameworks were built on Java and its cross-platform promise, but now new languages and technologies have emerged (e.g., Rails, AJAX, etc.) that are proving to be popular (or at least, trendy), making Java-based frameworks less desirable by some.

These issues are not necessarily unique to our work—indeed, the software engineering community has faced (and probably continues to face) similar issues. On the one hand, perhaps we need to take stock of research on object-oriented languages, component libraries, and end-user programming tools for guidance. But on the other hand, perhaps we also need to consider broader, more holistic supports for interaction design rather than just software integration to address the problem of facilitating the development of innovative learning technologies.

### 2. ESCOT and WILDs - Jeremy Roschelle & Roy Pea

As part of our drive for the Learning Sciences to have a larger scale impact in the everyday practices of math and science teachers, we have worked on re-use and interoperability issues for over 10 years. The ESCOT project, which used Java to interlink three research based technologies (The Geometer's Sketchpad, SimCalc, and AgentSheets) with supporting widgets, was perhaps the first major proof of concept that reuse and interoperability could work in educational technologies. In the WILD and G1:1 projects, we sought to understand how a community could collaboratively produce content for emerging wireless handheld platforms. While ESCOT did succeed in producing a series of "problems of the week" using interoperable and reusable technology that are still used by teachers today, no one is building on our platform – perhaps due to the immaturity of Java at the time of the project. Our handheld technologies also encountered a major platform issue: that handhelds have even MORE divergent operating systems than do desktop computers and its hard to get the community to agree on which platform to focus on. We are currently in better shape today, as the web-top platform (Web 2.0, Flash, Java) is stabilizing.

At present, we are investigating the form for a digital mathematics textbook that could aggregate improvements from the community without violating necessary properties of coherence, focus, and closure. For a digital mathematics textbook, the major challenge is to maintain coherence, focus and closure. A rare level of research-based consensus in mathematics education agrees that focus, coherence, and closure are key properties for accelerating learning. Our framework must allow a distributed team of editors to organize a process of improving a digital text while maintaining these fundamental properties. What tools will allow them to do that? We are building community for this effort, reaching out to organizations and teams with similar interests.

## 3. WISE, TELS and SAIL - Turadg Aleahmad & Jim Slotta

Re-use and interoperability have been themes in our software development practice over an 12-year time span in the WISE and TELS projects. WISE provided one of the first of its kind: An authorware environment where researchers could create their own interactive learning materials, then implement them in classrooms with no other software installation but a Web browser on all computers. WISE offers more than a dozen learning tools (graphing and datasets, drawing, journals, etc) that authors can configure. All student data is collected automatically via the Internet and provided to researchers and teachers in a user friendly portal. WISE has supported more than a dozen researchers around the world and is translated into six different languages. While some researchers have built new features into WISE, others have found it too restrictive. Later, we adapted WISE to serve a semester-long university curricuum in introductory computer science. While it was expandable into this new context, it took more effort than we expected and we learned many lessons and use cases in developing re-usable software for learning activities.

Since 2003, researchers and technologists from the Technology Enhanced Learning in Science (TELS) center have collaborated with colleagues internationally to create the Scalable Architecture for Interactive Learning (SAIL) - an open source java-based platform that can allow researchers and their technology specialists to create scalable, interoperable materials. The goal of SAIL is to enable software that could be re-used within a wide spectrum of research in the learning sciences. In designing our own applications, we separated out the “plumbing” so that it could be used in other applications, and in a way that we hope will support many different applications. While there has been some adoption of various portions of our software by other research labs, we are still in the early stages of establishing an open source developer community. In the years that we have spent developing SAIL, the most valuable part of our collaboration has been the knowledge community of software developers whose purpose is to support research in the learning sciences. We are now turning our efforts to supporting and expanding this community of technologists, to interface with researchers and to help them share code and expertise with each other.

## 4. Science Created by You (SCY) - Wouter van Joolingen & Ulrich Hoppe

Science Created by You (SCY) is an EU-funded Integrated Project that will deliver a system for constructive and productive learning of science and technology. SCY uses a flexible and adaptive pedagogical approach to learning based on "emerging learning objects" (ELOs) that are created by learners. In SCY-Lab (the SCY learning system) students work individually and collaboratively on "missions" which are guided by a general socio-scientific question (for example "how can build a climate-friendly house?") and fulfilling the mission requires a combination of knowledge from different domains (e.g., physics and mathematics, or biology and engineering). Students encounter multiple resources, collaborate with varying coalitions of peers, and use changing constellations of tools and scaffolds (e.g., to design a plan, to state a hypothesis etc.). The configuration of SCY-Lab is adaptive to the actual learning situation, advising students on appropriate learning actions, resources, tools and scaffolds, or peer learners that can support the learning process. The SCY approach is enabled by the innovative architecture of SCY-Lab that supports the creation, manipulation, and sharing of ELOs (models, data sets, designs, plans, etc.). SCY uses a Service Oriented Architecture, offering services for storage and retrieval of ELOs, collaboration, and management of learner profiles. Pedagogical decisions are based on information generated by pedagogical agents that exploit techniques of educational data mining to monitor information in the SCY repository that stores the ELOs as well as domain information, the log-files of student behavior, and the recorded chats between students.

 