Editor: **Jane Cleland-Huang**
DePaul University
jhuang@cs.depaul.edu

Toward Meaningful Industrial–Academic Partnerships

Jane Cleland-Huang

OVER THE PAST few years, there has been much discussion in the requirements engineering (RE) research community about researchers' responsibility to deliver practical, usable solutions to industry. Practitioners complain that researchers often work on theoretical problems that never seem to see the light of day, while the day-to-day problems remain unsolved. Nevertheless, many researchers collaborate closely with industrial partners and care deeply about technology transfer. Unfortunately, we face many hurdles in moving a viable research idea from inception to deployment.

> Many researchers collaborate closely with industrial partners and care deeply about technology transfer.

To encourage and foster technology transfer in the RE community, Daniela Damian and I initiated Ready-Set-Transfer, a game-show-style panel, at the 2011 IEEE International Requirements Engineering Conference. We designed it to encourage an honest exchange of ideas between researchers and practitioners. Instead of just talking about technology transfer, researchers present their ideas to a team of industrial judges, explaining the industrial motivation and describing how they've evaluated their idea's adoption readiness.

We've been running Ready-Set-Transfer each year since 2011. During that time, several interesting projects have emerged, representing ideas as varied as crowdsourcing, adaptive privacy, and traceability of safety-critical systems. This year, Jane Huffman Hayes and Didar Zowghi hosted the panel.[1] The three research projects they selected represented solutions from the diverse areas of requirements elicitation, modeling, and traceability. You can decide for yourself whether these solutions are successful in terms of technology transfer. Each one is at a different level of adoption readiness and carries a different degree of risk. I hope that talking about them will help us start thinking seriously about the issues, challenges, and responsibilities of technology transfer in RE.

This year, our panelists included Mike Panis from Teredyne (US), Juha Savolainen from Danfoss Power Electronics (Denmark), and Erik Bjernulf from Tolpagorni Product Management AB (Sweden) (see Figure 1). As you'll see, they made insightful comments to the contestants. Now, let's take a quick look at the three projects.

FlexiSketch

FlexiSketch (see Figures 2a and 3) is a unique collaborative sketching tool that University of Zurich researchers developed.[2] FlexiSketch runs across multiple tablets mimicking a distributed whiteboard. It supports free-form drawings and arbitrary node-and-edge diagrams. In addition, it aims to fill the gap between formal modeling and free-form drawing tools. Users can incrementally transform informal whiteboard sketches into formal models by retroactively creating new types using

 0740-7459/15/$31.00 © 2015 IEEE

a lightweight metamodeling feature. For example, they can select a shape on the whiteboard and promote it to a formal, reusable type in the metamodel. So, FlexiSketch supports a seamless transition from sketching to modeling. Whiteboard sketches become more than pretty pictures; users can evolve them into semiformal models that they can open in modeling-tool environments.

The FlexiSketch team have made a great effort to overcome the technology transfer hurdle, including uploading an early version to Google Play and encouraging requirements engineers from various companies to assist them in the early prototyping. They've also presented FlexiSketch at conferences attended by potential industrial adopters. Finally, they conducted three in-depth workshops at select companies to evaluate their approach and investigate people's sketching and notation-defining behavior when using FlexiSketch.

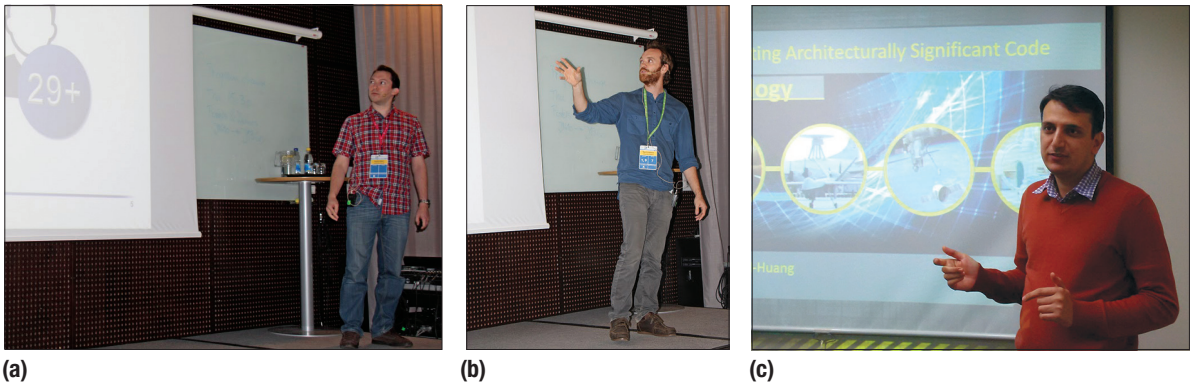
University of Zurich researcher Dustin Wüest said this experience has taught them that platforms and technologies matter. Several companies were willing to try FlexiSketch only if an iOS version was available. Dustin also pointed out that research and industry needs often diverge. Whereas researchers tend to focus on proposing and evaluating novel ideas and concepts, industrial adopters want rich, stable feature sets, which can be costly to deliver. In addition, Dustin mentioned that taking a research idea to market requires a team effort that involves not just scientific discovery but also outreach and marketing skills.

FIGURE 1. The panelists from the 2014 Ready-Set-Transfer panel were Mike Panis (Teredyne, US), Juha Savolainen (Danfoss Power Electronics, Denmark), and Erik Bjernulf (Tolpagorni Product Management AB, Sweden).

The industrial panel asked how the team could be certain that early adopters were using the novel features of the semiformal metamodeling and not just adopting FlexiSketch for its basic collaborative-whiteboard features. The team answered that they already had been addressing this through the face-to-face industrial workshops they had conducted, which gave them deep insights into the features people were using. Indeed, FlexiSketch is much more than a collaborative-whiteboard sketching tool.

Creativity for Requirements Discovery

Martin Mahaux from Namur University has developed a slew of creativity solutions for requirements



(a) (b) (c)

FIGURE 2. Presentations of projects shown at Ready-Set-Transfer in 2014. (a) Dustin Wüest presenting FlexiSketch. (b) Martin Mahaux presenting the Collaborative Creativity Canvas. (c) Mehdi Mirakhorli presenting Archie (this photo was taken at another event).

FIGURE 3. Martin Glinz, Norbert Seyff, Dustin Wüest, and Parisa Ghazi work with FlexiSketch. For additional information, visit www.ifi.uzh.ch/rerg/research/flexiblemodeling.html.

FIGURE 4. Participants in one of Martin Mahaux's workshops use the Collaborative Creativity Canvas to explore innovative requirements ideas. For more details, see http://bit.ly/martinmahaux.

discovery.[3] One such solution is the Collaborative Creativity Canvas (see Figures 2b and 4). Facilitators can use it to foster creativity and replace the often frustrating requirements negotiation process with a lively cocreation process. It aims to turn stakeholder conflicts into opportunities for innovation.

Although Martin's research included traditional literature reviews, expert opinion, and practitioner surveys, it was driven largely by industrial collaboration. Ideas conceived in industry were iteratively and incrementally improved as they moved back and forth between the lab and practice. As such, Martin's creativity solutions resulted from industrial partnerships and not through the more traditional model in which an idea emerges from research and then incubates in a lab for five years before the finished product is offered to industry.

Martin explained that this project revealed the benefits of industrial codesign, especially through the ongoing guidance and feedback he obtained. He pointed out that working with industry didn't produce short-sighted research because he had the time and freedom to consider, and explore, innovative ideas throughout the process.

Archie

Architectural knowledge and related quality concerns are often undocumented and tacit. So, developers often lose track of early design decisions. For example, system-level qualities representing "nonfunctional" requirements tend to become eroded during refactoring, bug fixing, and other maintenance activities.

To address this problem, my research group at DePaul University developed Archie (see Figures 2c and 5), an Eclipse plug-in.[4] Archie focuses on requirements' role in a project's downstream design and maintenance phases. It parses source code and then automatically detects and visualizes a range of architectural tactics such as heartbeats, resource pooling, and role-based access control.

Archie was funded by grants from the US National Science Foundation and Department of Homeland Security and developed by Ahmed Fahkry (see Figure 6) and other students under Mehdi Mirakhorli's supervision. To place Archie into practitioners' hands, we released it on GitHub under Archie-Smart-IDE and on the Department of Homeland Security's SWAMP (Software Assurance Marketplace).

Mirakhorli explained that one of the greatest challenges for technology transfer was in understanding the real users' actual usage patterns. We addressed this through frequent iterations of prototyping, coding, and testing. However, the real test will come as industrial users adopt Archie in their development environments. As such, Archie is less advanced along the technology-transfer scale than FlexiSketch or Martin's creative collaboration activities.

The panelists questioned whether developers on real projects would find the prepackaged set of trained classifiers fit for their purpose. The Archie presenters explained that Archie was highly flexible and that teams could construct new tactic templates and either train new classifiers or use Archie's click-and-point features to map code sections manually to the new tactics.

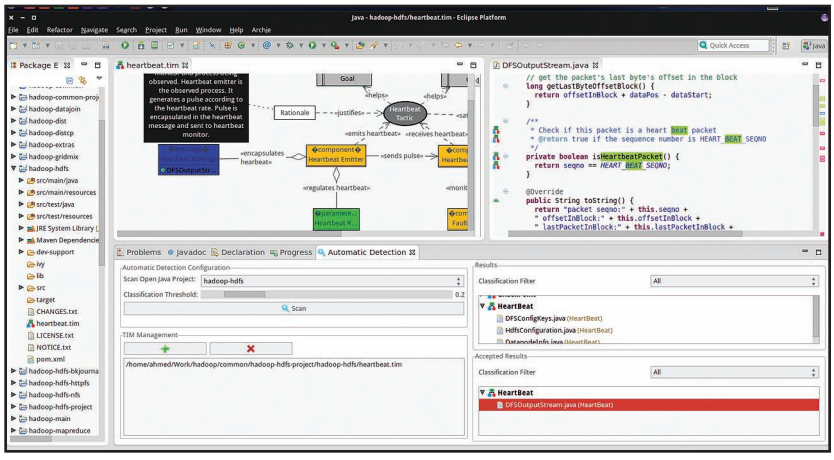


FIGURE 5. A snapshot of Archie showing the detected heartbeat tactic highlighted in code and visualized in graphical form. For a demo of Archie, visit http://re.cs.depaul.edu/mehdi/Archie.mp4.

What's Next?

The Ready-Set-Transfer panel always raises interesting technology-transfer issues—some of which we have answers for and some of which we don't. Mike Panis saw benefits to both sides and explained that the panel "helps researchers—regardless of whether they are contestants or in the audience, to step back from the potential, future value of their research and consider what would be needed for it to provide immediate benefits." He also observed that "it helps practitioners consider whether they can apply research results to their current work."

So, challenges and opportunities abound. In my research, I've found that the biggest adoption barriers are the cost and effort of bringing viable research prototypes to industrial standards. A typical research grant doesn't normally include funding for this kind of technology transfer, so researchers must proactively seek additional funding to jump the readiness hurdle.

One thing is clear. Successful technology transfer needs both sides of the partnership. We can't succeed unless researchers and practitioners work together to address important problems that a typical software development project can't accommodate. Neither can we be confident that we're addressing the right problems at the appropriate scale and complexity without industry's feedback and willingness to share data and expose its challenges.

As always, I'd love to hear from you. In a future column, I'd like to give more voice to practitioners. So, I invite you to email me and tell me about problems you're experiencing that you wish researchers would address or about your success or failure stories regarding technology transfer. Let's engage in an ongoing, fruitful discussion so that we can see innovative requirements projects—possibly seeded from industry—make their way to industry as viable, effective solutions.

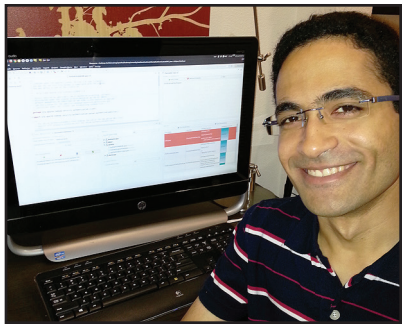


FIGURE 6. Ahmed Fahkry, Archie's lead developer, explained that working on Archie gave him the opportunity to engage in a challenging research project as a graduate student.

References

1. J.H. Hayes and D. Zowghi, "Ready-Set-Transfer! Technology Transfer in the Requirements Engineering Domain" (panel discussion), *Proc. 19th IEEE Int'l Requirements Eng. Conf.* (RE 14), 2014, pp. 500–501.
2. D. Wüest, N. Seyff, and M. Glinz, "Semi-automatic Generation of Metamodels from Model Sketches," *Proc. 28th IEEE/ACM Int'l Conf. Automated Software Eng.* (ASE 13), 2013, pp. 664–669.
3. M. Mahaux et al., "Collaborative Creativity in Requirements Engineering: Analysis and Practical Advice," *Proc. 2013 IEEE Int'l Conf. Research Challenges in Information Science* (RCIS 13), 2013, pp. 1–10.
4. M. Mirakhorli et al., "Archie: A Tool for Detecting, Monitoring, and Preserving Architecturally Significant Code," to be published in *Proc. 22nd ACM SIGSOFT Int'l Symp. Foundations of Software Eng.* (FSE 14), 2014.

JANE CLELAND-HUANG is a professor of software engineering at DePaul University. Contact her at jhuang@cs.depaul.edu.

See www.computer.org/software-multimedia for multimedia content related to this article.