Editor: **Sarah Gregory**
Intel Corporation
sarah.c.gregory@ieee.org

The Unplanned Journey of a Requirements Engineer in Industry

An Introduction

Sarah Gregory

IN NEARLY 17 years as a requirements engineer, I've yet to meet anyone in industry in the US who went to university with that profession as his or her intended career. (Yes, yes, I know there are exceptions to any general statement. I'll save you the trouble of looking for my email to send me a rebuttal or your own story of becoming a requirements engineer—it's sarah.c.gregory@ieee.org.) Instead, many of us landed in the discipline by chance or accident, usually without adequate preparation. Any requirements engineer will, of course, be quick to note that "adequate" is itself a weak and ambiguous word.

Some of us who begin to write requirements find that we love the practice and decide to focus on it, and claim the title of "requirements engineer" as our own. Many others whose job involves writing requirements work in other disciplines but use requirements engineering (RE) techniques in some of their day-to-day work. My own origin story—how I ended up in RE—not only informs the work I do now but also drives questions about the discipline I hope to explore with you as I begin my term as the Requirements department editor.

Surprised by RE

The height of the dot-com boom seemed like the perfect time to jump out of legal academia and into tech, or so I thought when I resigned my academic position as a legal reference and electronic services librarian. I closed my office door and took my newly minted information science degree with its emphasis on human–computer interaction to a cubicle at Intel.

My first position at Intel involved usability design, analysis, and testing for home-networking devices. I was tasked with crafting an installation sequence for a new DSL modem. I turned to one of the program leads and said, "I'll get started on that now. Can you point me to where I can find the requirements?" An uncomfortable silence followed. "Well, we were hoping that you might be able to write those, too," he said.

Thus began my career in RE—with requirements scrambling. I wandered from cubicle to cubicle until I found a network engineer who shared a template he had used for requirements on a previous project. Armed with no experience and a profound fear of failure, I set off to find the people and information I'd need to fill in the blank boxes in that document.

0740-7459/17/$33.00 © 2017 IEEE

At the time, I had never heard of the discipline of requirements engineering. My graduate-school program hadn't taught it, and my business unit had no self-identified "requirements engineers." Instead, many project team members wrote requirements in their own areas of expertise—for example, network protocols, industrial design, or thermal or mechanical systems—but few claimed expertise in RE itself.

A new group in our corporate quality organization aimed to address that gap. Erik Simmons, the newly hired requirements lead, was due to offer a Writing Good Requirements course through Intel's in-house training program. I took the class as soon as it was available, desperate to learn enough to fill in that template with confidence, if not quite authority. Armed with only a day's worth of education, I was able to go back to the template and begin to engineer a few requirements, rather than just scramble for information. Progress was made.

When a reorganization shifted me into a group closer to Intel's core silicon business, a program manager for a huge business-process-and-tool-reengineering project selected me to lead the requirements effort. My qualification? I was the only person on the team who had taken Writing Good Requirements. So, with Erik's support and mentoring, I soon found myself teaching the same seven-hour seminar I'd taken but a few months earlier. I helped dozens of others—fabrication techs, silicon designers, new-product planners, software developers, and more—write their requirements for the program, using a powerful but simple set of practices drawn from Tom Gilb's *Competitive Engineering*.[1] Over the course of that program my professional identity shifted. I was no longer a usability analyst and tech writer; instead, I became a requirements engineer. That class is currently in its eighth major edition and continues to be taught regularly at Intel.

Corporate data retention policies and hard drive size limits restrict the amount of previous work that remains in my archives. I do have some of those early requirements, though, and every so often I go back and look at the specifications I wrote when I knew I needed to write requirements but had little understanding of the art and science of crafting good ones. When I teach the current version of that training class—something I've been able to do about once a month and around the world—the content on the slides supplements the lessons I've learned from years of hands-on work with so many varied teams and individuals. Sometimes I've been the author of the requirements, sometimes a reviewer, sometimes a coach or mentor, and sometimes just another person in the

TABLE 1

EARS (Easy Approach to Requirements Syntax) patterns cover specific types of functional requirements and constraints.

Pattern name	Pattern
Ubiquitous	The <system or actor> shall <action> <object>
Event-Driven	WHEN <trigger> <optional precondition> the <system or actor> shall <action> <object>
State-Driven	WHILE <system state or actor state>, the <system or actor> shall <action> <object>
Unwanted Behavior	IF <unwanted state or unwanted event>, THEN the <system or actor> shall <action> <object>
Optional Feature	WHERE <feature is included>, the <system or actor> shall <action> <object>
Compound	Combinations of the previous patterns

room or on the phone trying to muddle through a drawing or paragraph with someone who found my name in the corporate directory.

Some days are easier than others, but I've not faced a single day of work as a requirements engineer that has been boring or where I've felt I know everything I need to do my job. Indeed, as I look back to my early days working with a template in which I thought I needed to just fill in the blanks, I'm more aware than I was back then of how much I still want to learn.

A Particular Source of Support

In my tenure as a requirements engineer, without a doubt my best professional development has occurred during one week in late summer or early autumn for the last 10 years: the annual IEEE International Requirements Engineering Conference (requirements-engineering.org).

For several years before 2008, Erik Simmons and occasionally another RE colleague or two attended that conference and brought back articles to read, ideas to test in our own environment, and methods that might help us improve our company's RE practice. In 2008, he was asked to serve on a conference panel but couldn't, so I attended in his place. With no small amount of trepidation, I flew to Europe to represent my company in a discussion of industrial RE practice. I was convinced that my abject ignorance would be on full display and that every other company had mastered what we struggled to do.

To my surprise, over the course of the panel discussion "Industry Issues in RE," I learned not only that others had suggestions for some of the challenges I experienced in my practice but also that I could offer a few ideas to my industry colleagues. Some researchers were exploring topics I didn't understand and certainly couldn't begin to apply to my practice. Others were working on issues we'd struggled with ourselves—for example, traceability or writing requirements to meet regulatory and safety standards.

Some conference presentations have been particularly memorable and helpful. At RE 2009, Alistair Mavin of Rolls Royce delivered a presentation that gave us the idea to pilot the basic constrained natural-language sentence patterns called EARS (Easy Approach to Requirements Syntax; see Table 1).[2] EARS is now a standard part of our in-house training. In 2011, Frank Houdek presented "Semi-automatic Identification of Features in Requirements Specifications."[3] Mike Panis's 2010 case study "Successful Deployment of Requirements Traceability in a Commercial Engineering Organization … Really"[4] is a reference I still use when I'm working with teams on hierarchical decomposition and tracing of requirements. That paper drew from Olly Gotel and Anthony Finkelstein's research on traceability[5] and pointed to the work of Jane Cleland-Huang, my predecessor in this column.[6]

Through engagement with the community formed through the RE conference, I've had the privilege to meet some of the authors of those articles and books to which I clung for dear life as I built a career in a field I'd not heard of until I was dropped in the midst of it. Over breaks, meals, and late-night discussions at the annual RE conferences and other gatherings, my RE colleagues and I have explored ideas, discussed new work from researchers and industrial partners, and debated the future direction of the practice. We've picked up ideas from papers and proceedings and have tested a few. Some have stuck; others haven't worked at all in our complex, wildly varied contexts. However, our understanding and practice have improved simply from increasing the depth of our understanding.

So it's with gratitude and admiration, and quite aware of the task before me, that I put my fingers to the keyboard following Jane Cleland-Huang's tenure as the Requirements editor. Those of us working in RE in industry are so often the gleaners of the profession. Jane and our other research colleagues theorize, strategize, experiment, publish, and present. On occasion, we in industry express our frustration that their areas of focus don't solve the critical problem we need fixed last week, thanks. (No, I can't tell you about it—intellectual property. Sorry.) They continue to drive forward and outward, and even if we can't always talk about the challenges we experience, we learn from their publications and presentations.

But over the past 40 years, since the inception of RE as a discipline, industry and research dialogue has been essential to advancing quality practice in our companies. I'd also like to think that our participation in the community helps inform our research friends of the next generation of wicked problems that will need to be solved and where we sorely need their input. The boundary between research and industry seems especially permeable in our field, and opportunities for information sharing abound.

With this column, I hope to explore current issues faced by industry and highlight new work from our research counterparts that might otherwise escape notice by those of us who focus primarily on our companies' products. Although I've shared some of my professional history in this introduction, my columns won't be about my company's work. Instead, I'll discuss the common and uncommon experiences of RE in practice, across a range of industrial contexts.

Industry colleagues, I invite you to share your experiences with RE; especially consider contributing a guest column in 2018. What RE challenges keep you up at night? RE is 40 years old in 2017; where do you see the practice headed over the next decade? Researchers, what are you working on now that industry needs to consider over the next three to 10 years? I look forward to our discussions.

References

1. T. Gilb, *Competitive Engineering: A Handbook for Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage*, Butterworth-Heinemann, 2005.
2. A. Mavin et al., "Easy Approach to Requirements Syntax (EARS)," *Proc. 17th IEEE Int'l Requirements Eng. Conf.* (RE 09), 2009, pp. 317–322.
3. E. Boutkova and F. Houdek, "Semi-automatic Identification of Features in Requirement Specifications," *Proc. 19th IEEE Int'l Requirements Eng. Conf.* (RE 11), 2011, pp. 313–318.
4. M.C. Panis, "Successful Deployment of Requirements Traceability in a Commercial Engineering Organization ... Really," *Proc. 18th IEEE Int'l Requirements Eng. Conf.* (RE 10), 2010, pp. 303–307.
5. O.C.Z. Gotel and A.C.W. Finkelstein, "An Analysis of the Requirements Traceability Problem," *Proc. 1st Int'l Conf. Requirements Eng.* (ICRE 94), 1994, pp. 94–101.
6. J. Cleland-Huang, "Just Enough Requirements Traceability," *Proc. 30th Ann. Int'l Computer Software and Applications Conf.* (COMPSAC 06), vol. 1, 2006, pp. 41–42.

SARAH GREGORY is a staff requirements engineer at Intel Corporation. Contact her at sarah.c.gregory@ieee.org.