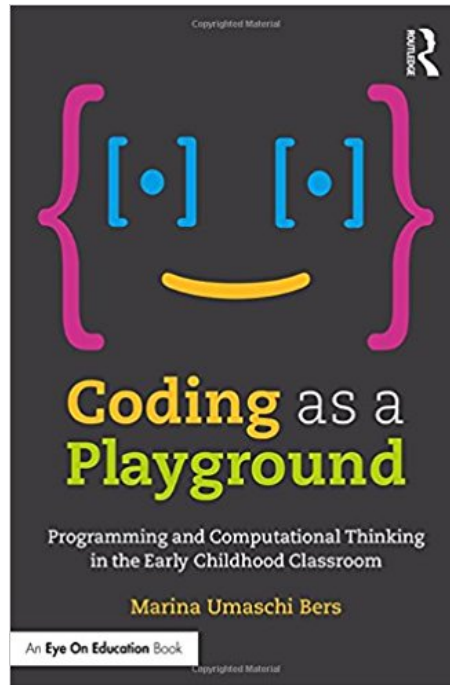


Interview with Professor Marina Umaschi Bers

We recently caught up with Professor Bers from Tufts University about her new book *Coding As a Playground*.



Q: What is *Coding as a Playground* about?

A: The book is about how to bring new technologies that allow children to create stories, to create games, and to become programmers, and to do it in a fun playful way. The book describes this approach to introducing new technologies that teach children how to program either on the screen or using a robotic system--tangible ways--but in a fun, playful, developmentally appropriate way and that means pretty much what children do in a playground. When children are in a playground, they are engaging in many domains of development: personal development, physical development, social development, emotional development, cognitive development, even moral development. When children are engaging with coding, we want them to also engage in those domains.

Q: So you're not using the word playground just as a metaphor. You mean it to be like an actual playground when they're coding?

A: Yes, yes. It's not just a metaphor it actually is about engaging in similar positive behaviors as the ones that we have in the playground: talking with other people, interacting socially, creating, inventing, discovering, playing, learning to solve conflicts, running around, experimenting. But the playground also serves as a metaphor because, when thinking about technologies as

playgrounds, I'm contrasting the *playground* with the *playpen*. Playpens are very limited, very isolating, very few choices. Playgrounds, instead, are open-ended and offer lots of opportunities for creativity. So the book really focuses on the latter, coding as a playground type of experience.

Q: A playground has a set number of items (swings, slides, seesaws), but when a child is on a playground, each time it's a new experience. It isn't the implements that are in the playground but how the child interacts with them. Is that correct?

A: That's exactly it. When you engage children in coding environments--for example in the book we describe what we did in creating Scratch Jr and Kibo robotics--we provide a certain set of structures (programming blocks, sensors, etc.), and the child can play with them and can use them in very, very different ways and it never gets boring. Every time they code they make something else, same thing as a playground: You have swings and seesaws, but every time you go the experience is different.

In a sense we could think of books as playgrounds. Sometimes we keep reading and reading the same book over and over. Every time we do it the experience is different. The structures in a playground, or the objects in the computer programming environment, are the same but the meaning we make with them is completely different.

Q: Who is the target audience for this book?

A: The book is for adults not for children and it's for any adult--parents, teachers, informal educators, museum educators, pediatricians--anyone that's in contact with young children, and by young children I mean ages 4 to 7, and are interested in introducing technology into the lives of children and are wondering how to do it in a developmentally appropriate way. Most specifically, if you are a librarian or teacher who is working with technology in schools, you would really benefit, but in general this book is not written in academic jargon and it is written for the general public and for those who want to understand the potential of engaging young children in learning how to program and how to use robotics in developmentally appropriate ways.

Q: Can you talk about parents as the target audience?

A: The book comes with a lot of stories about the kinds of things that children can do when creating a computer program and a robot. For example the book describes many projects with Scratch Jr., which is a free app we have developed in my DevTech research group at Tufts University, in collaboration with Mitch Resnick from the MIT Media Lab. Any parent can download Scratch Jr. at no cost and have their children program the kinds of things I describe in the book: animated stories, interactive games, art projects. The book also describes multiple experiences in which the KIBO robot that we developed in my DevTech group, which allows children to program using wooden blocks, without screens or keyboards, was used to tell stories and explore engineering, programming, and literacy in an integrated way. I think the book

provides a window into understanding what kinds of technologies are playgrounds and what kinds of technologies are playpens. Technologies are in the life of children and households and educating everyone about how to choose a technology is helpful.

We hear so much about STEM programs and the word STEM has become very popular, and this book says, Wait a minute what are we really talking about? Let's break it down by letters. This book is focusing on the "T" of technology. But most specifically on activities with technology that are playground-like activities, such as learning how to code, making robots, which is helping unpack the T of technology. One of the strengths of the book is that it isn't really about STEM per se; it's really coding tied to literacy, having to do with language as a form of expression that also engages in problem solving.



Q: How can reading teachers benefit from your book?

A: In the book, because I draw very heavily on models of literacy, I show how coding is a form of expression, and at the end of the day that's the goal of teaching kids how to read and write, so they can express themselves. In very concrete ways, for example, you can find many classrooms with Scratch Jr. or the Kibo robot with teachers and kids sitting in a circle and reading a book, and children are invited to come up with a different ending for the book. Instead of writing it down they can create a play using, for example, the robot and they can program the different Kibo robots to behave as different characters in the book so the book can come alive through a robotic dramatic play. Or through Scratch Jr. they can animate the book or come up with different endings or different stories. When we work with coding we usually work with books as well.

Q: It sounds like you constructed not just the book but Scratch Jr. and that whole experience to integrate organically with the early classroom environment. Is that correct?

A: Yes, it is very much organically integrated, and it provides, it augments, the experience of literacy because it provides kids with an opportunity to put a story into a different format (play, etc.).

Q: What sort of PD does a teacher need to use *Coding As a Playground*?

A: At Tufts University, we do offer different kinds of PD. We have short two-day programs during the summer, as well as a year-long blended online/face-to-face graduate certificate program in Early Childhood Technology. We also do short two-hour programs. Most early childhood teachers have never done any programming or any robotics. The minute they encounter the technology, they realize it's pretty easy and they know how to integrate it into their best practices that include the teaching of literacy through playful approaches.

Our technologies, KIBO and Scratch Jr, were designed for very young kids who can't read or write. So a teacher can understand the technology pretty easily. We call our approach "Coding as Another Language." Teachers who are teaching language can easily learn how to do it. We believe coding should be taught integrated into whatever subject is already happening in early childhood, including art, and movement, and in the playground. Everything.

Q: What do you mean by coding as a type of literacy?

A: The first thing I want to make sure we understand is that I'm not advocating at all ~~the~~ replacing of the teaching of a language with the teaching of coding. There's room for teaching English, teaching Spanish, teaching French, etc. What I mean by literacy is something else. What I mean by literacy is the understanding of how to manipulate and understand a symbol system. An alphabet is a symbol system. A programming language is also a symbol system: It has its syntax, semantics, and pragmatics, and it has its way of combining things. When we do computer science, that way of combining things to make a set of instructions for something to happen is called an algorithm. The equivalent in English and Spanish we call a sentence. Then there are algorithms that are a bit more complex in the same way sentences can be combined to create a paragraph. And paragraphs can be combined to become chapters in a story. The same is true for algorithms. We have an abstract system that includes the symbolization and representation in which logic organizes its elements. We need to use that logic to combine, and all of that put together allows us to express ourselves, to communicate ideas. We can do generation of ideas and also comprehension of ideas. In the same way that we can debug our programs, we can edit our sentences in our rough drafts. And over time, the more we use it, we become fluent with that other language: a natural language or a programming language.

Q: Tell us about the Scratch Jr. programming language.

A: Scratch Jr. is a free app I developed with Mitch Resnick, along with the Playful Invention Company. Through a grant from the National Science Foundation and the support of the Scratch Foundation, we developed this free programming language for kids aged 5 to 7, but we have been working with even younger kids, 4 to 7. It's icon based and it really comes from some of the ideas developed from Scratch, which is a programming language for older kids. But what happens is that most young kids at the time they are first learning how to read and write they can't use Scratch. It's too complicated. It's too sophisticated. So we decided to create a programming language for very young children in a language they can understand. That's how we started Scratch Jr.

There's a palette of commands. It's very colorful. There are very few words, although if you put your finger over a command the words are displayed. We are very conscious that we want kids, even if they can't read at the time, we want them exposed to letters and words. Just like in a kindergarten classroom where objects are labeled.

Kids can program characters, they can design backgrounds, and they can create stories. They have the ability to create sequences through pages and make stories. They can do anything they want, and it has become very popular. We have over nine and half million downloads from every single country in the world, except North Korea and Western Sahara. We've had it translated into any language you can think of. It's great. It works on tablets, on Android and iOS devices, on the desktop. And it is free.

Q: What resources are available to using Scratch Jr. in the classroom?

A: If you go to scratchjr.org, we have curriculum that teachers can download. Some of it is explicitly integrated with literacy and writing, while others are for games, and others for math. We have also published a book called *The Official Scratch Jr. Book*. It includes tons of activities that parents can do at home. This book is also for teachers, but parents seem to like it because it's not tied to a curriculum.

And we have big news. In June we are launching a set of Scratch Jr. Coding Cards. This is a set of 75 cards for introducing different concepts from Scratch Jr. This is for cases where you don't have time for a curriculum. If you don't have 20 hours or even 10 hours, but only 15 minutes--what can you do? You can use these Scratch Jr. Coding Cards.

Q: Finally, what advice do you have for children to become better readers?

A: I'm not a reading teacher or a literacy teacher, but I am a lover of reading. My advice is to read with your children. Read, read, read, read, read with them! Read things that are interesting to them. It doesn't matter the level of the book. When you read with your child, read something that she or he is really interested in, in learning about, in immersing him- or herself about. If there are some hard words, explain the hard words, and if you don't want to stop the story to explain, keep going. Let the children fall in love with the story.

About Professor Marina Umaschi Bers

Marina Umaschi Bers is a professor at the [Eliot-Pearson Department of Child Study and Human Development](#) and an adjunct professor in the Computer Science Department at Tufts University. She heads the interdisciplinary [Developmental Technologies](#) research group. Her research involves the design and study of innovative learning technologies to promote children's positive

development. She also developed and serves as director of the graduate certificate program on [Early Childhood Technology](#) at Tufts University.

Prof. Umaschi Bers is from Argentina, where she did her undergraduate studies in Social Communication at [Buenos Aires University](#). In 1994 she came to the US where she received a Master's degree in Educational Media and Technology from [Boston University](#) and a Master of Science and PhD from the [MIT Media Laboratory](#) working with Seymour Papert.

Prof. Umaschi Bers received prestigious awards such as the 2005 Presidential Early Career Award for Scientists and Engineers (PECASE), the highest honor given by the U.S. government to outstanding investigators at the early stages of their careers, a [National Science Foundation \(NSF\)](#)'s Young Investigator's Career Award, for her work on virtual communities of learning and care, and the [American Educational Research Association \(AERA\)](#) Jan Hawkins Award for Early Career Contributions to Humanistic Research and Scholarship in Learning Technologies. She was also selected by the Boston Business Journal as one of the recipients of its 2015 Women to Watch in Science and Technology awards.

Prof. Bers also developed the [KIBO robot kit](#) for children 4 to 7 year old, that can be programmed with wooden blocks without using keyboards or screens old. To make KIBO widely available, she co-founded a start-up, [KinderLab Robotics, Inc](#) that with funding from the National Science Foundation SBIR program, is commercializing [KIBO](#) and making available worldwide. As of 2018, KIBO can be found in 52 countries and countries such as Singapore and Argentina have launched national programs including it.