
Tangible Kindergarten

Learning How to Program Robots in Early Childhood

Marina Umaschi Bers,
Eliot Pearson Department of Child Development,
Tufts University, Medford, Massachusetts

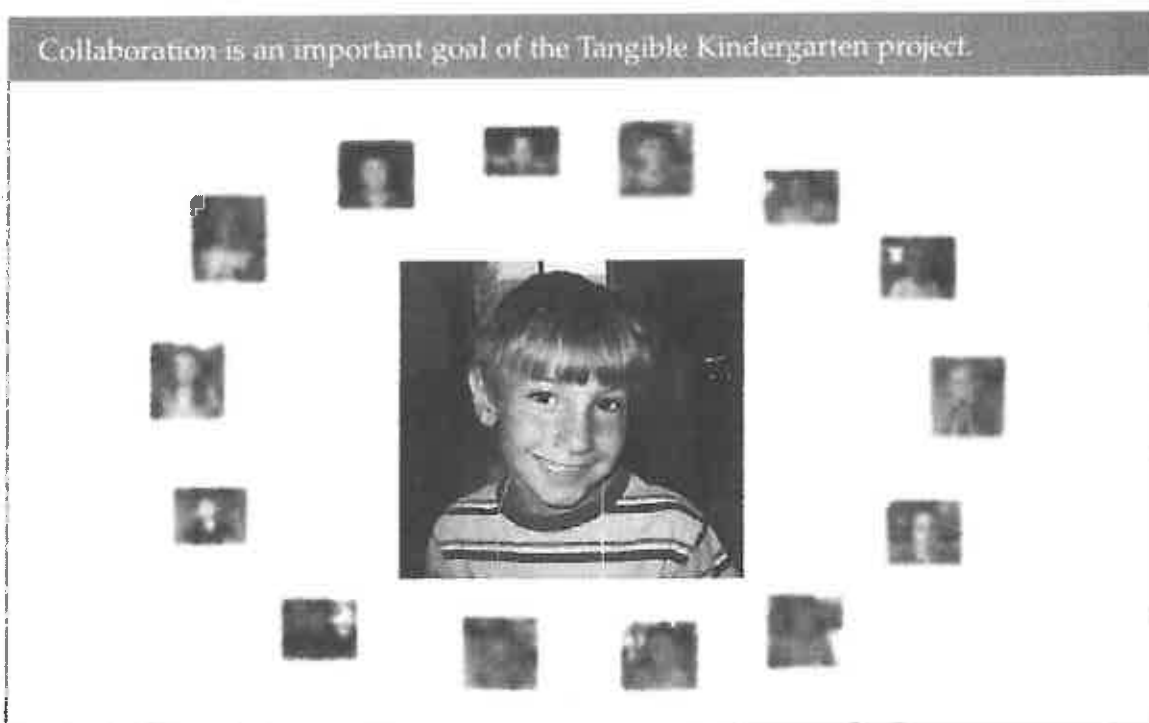


Image courtesy of Marina Bers.

As shown on the previous page, during Tangible Kindergarten® (Tangible K) classes each child receives a personalized printout with his or her photograph in the center of the page and the photographs and names of all other children in the class arranged in a circle. The children use this image to keep track of their collaborations with other children during the day so that at the end of the week they can write or draw “thank-you cards” to the children with whom they have collaborated.

Tangible K is a curriculum that promotes powerful ideas from computer science, in particular robotics, among children in Grades preK–2. Tangible K is specifically designed to engage young children in computational thinking and engineering design while integrating relevant concepts and skills with other areas of the early childhood curriculum. The curriculum promotes positive uses of technology and technological fluency. With an emphasis on the concept of sequencing as a core idea of computer programming that is relevant in the early years as a major predictor for academic success in literacy and math, the Tangible K curriculum also supports 21st century skills. Students engaged with the Tangible K curriculum work individually, in pairs, and in teams to program a robot’s behaviors. In the process, they apply knowledge of mathematics, use inquiry and problem-solving skills, and develop their creativity by using the engineering design process.

The Tangible K curriculum is interdisciplinary and specifically designed to address the developmental stage of children in Grades preK–2. It integrates powerful ideas, concepts, and skills from the fields of computer science and engineering with traditional curricular areas such as math, science, literacy, social sciences, and the arts. The research shows that the curriculum helps children develop engineering and computer science practices while they build and program their robots. It also shows that children increase their sequencing skills, a foundational concept that is a predictor of later academic success. Furthermore, the Tangible K curriculum was designed to promote positive uses of and attitudes toward technology.

To date (September 2013), the Tangible K curriculum has been used by more than 250 early childhood teachers and has reached more than 2,000 students.

Six Powerful Ideas

In order to engage children in computational thinking, the Tangible K curriculum focuses on six powerful ideas, communicated to young children through activities.

Robotics is the engineering discipline that focuses on the creation and programming of robots, which are machines that can follow instructions and move on their own to perform tasks.

Activity: *What Is a Robot?* After an introduction to robotics by looking at different robots and talking about the functions they serve, children build their own robotic vehicles and explore their parts and the instructions they can use to program them.

The engineering design process is used to develop products to solve a need or problem. It has several iterative steps: identifying a need or defining the problem, doing research, analyzing possible solutions, developing the product, and communicating and presenting the product.

Activity: *Sturdy Building*: Children build a nonrobotic vehicle to take small toy people from home to school. The vehicle needs to be sturdy as well as perform its intended functions.

Sequencing, control, and flow. A sequence of instructions can be described in a program and acted out in order by a robot. Each block has a specific meaning. The order of the blocks is important.

Activity: *The Hokey-Pokey*: Choose the appropriate commands and put them in order to program a robot to dance the Hokey-Pokey.

Loops and parameters. A sequence of instructions can be modified to occur over and over again. Control flow commands can be qualified with additional information. For example, loops can be modified to repeat forever or a concrete number of times.

Activity: *Again and Again Until I Say When*: Students use a pair of loop blocks ("repeat"/"end repeat") to make the robot go forward again and again, infinitely and then just the right number of times to arrive at a fixed location.

Sensors. A robot can use sensors, akin to human sense organs, to gather information from its environment. Sensor information can be used to control when the robot follows given commands.

Activity: *Through the Tunnel*: Children use light sensors and commands to program a robot to turn its lights on when its surroundings are dark and vice versa.

Branches. At a branch in the program, a robot can follow one set of commands or another depending on the state of a given condition.

Activity: *The Robot Decides*: Students program their robot to travel to one of two destinations based on light or touch sensor information.

Overview of the Tangible K Curriculum

The Tangible K curriculum is designed for a minimum of 20 hours of classroom work, divided into the following structured sessions based on the six powerful ideas identified earlier:

1. Sturdy Building (the engineering design process)
2. What Is a Robot? (robots have special parts to follow instruction)
3. Hokey-Pokey: sequence of commands (the sequence or order of commands matters)
4. Again and Again Until I Say When (loops and number parameters)
5. Through the Tunnel (sensors and loops)
6. The Robot Decides (sensors and branches)

The curriculum unit is designed to take place over the course of one intensive week of work or over the course of several months, with several short sessions per week. Depending on children's developmental levels and prior experience with digital technology, programming, and robotics, students might need more or less time. One issue for each teacher to resolve is how long to allot for each of the six lessons. Each can be spread out over several sessions to accommodate the classroom schedule and students' attention spans for this work.

Tangible K involves children in making their own projects (i.e., creating content) by engaging them in making a robotic artifact and in programming its behaviors. Following

is an overview of the six sessions. Themes vary in the pilot versions of this curriculum—transportation, community, animals—but the powerful ideas of computer science and robotics remain the same. The transportation curriculum is used here as an example.

Lesson 1. Sturdy Build (introduction to engineering design). Students build a sturdy, nonrobotic vehicle using LEGO bricks and other materials, and they use design journals to learn the engineering design process. As a result of this activity, students will understand that LEGO bricks and other materials can fit together to form sturdy structures, and the engineering design process is useful for planning and guiding the creation of artifacts.

Lesson 2. What Is a Robot? Students describe the components of a robot, including the brain, motors, and wires. They upload a program to a robot via the tangible blocks or computer interface and build a sturdy, robotic vehicle using LEGO bricks and other materials. Through these activities, the students come to understand that robots need moving parts, such as motors, to be able to perform behaviors specified by a program. The robotic brain has the programmed instructions that make the robot perform its behaviors; and it must communicate with the motors for them to function.

Lesson 3. Hokey-Pokey (sequence of commands). The students select the appropriate block corresponding to a planned robot action, then connect a series of blocks. The students then upload a program to the computer and transmit it to a robot. From these activities, students learn that each icon corresponds to a specific command; a program is a sequence of commands that is followed by a robot; and the order of the blocks dictates the order in which the robot executes the commands.

Lesson 4. Again and Again Until I Say When (loops and number parameters). The students recognize a situation that requires a program to use loops. They then write a program that loops and use parameters to modify the number of times a loop runs before the program stops. Students learn that a command or sequence of commands may be modified so that they repeat. Some programming commands, like “Repeat,” can be modified with additional information. Also, a simple program that uses fewer blocks is better than a complex one that accomplishes the same goal.

Lesson 5. Through the Tunnel (sensors and loops). Students connect a light or touch sensor to the correct port on the robot. They then write a program that includes waiting for a specific condition. They learn that a robot can “feel” and “see” its surroundings through the use of sensors, and a robot can react to collected data by changing its behavior. Also, a robot can be programmed to remain on a certain task until a specific condition is met.

Lesson 6. The Robot Decides (sensors and branches). The students connect a light or touch sensor to the correct port on the robot and identify a situation that calls for a branched program. They then write a program that uses a branch. They learn that a robot can “choose” between two sequences of commands depending on the state of a given condition.

Each session follows a similar format: (1) warm-up games to introduce the new concept or powerful idea in a playful way, (2) a building and/or programming task to reinforce the powerful idea underlying the lesson, (3) working on a small project (individually or in pairs) that uses the powerful idea in a new context, (4) participate in a technology circle to share learning process and products, and (5) assessment.

After the six Tangible K sessions, the class creates a final project focusing on a particular theme. This is an opportunity to revisit the learned concepts and skills, applying them to a project related to other curricular content. The length of time for these projects varies according to the group of students and the teachers' goals, expectations, and curricular demands. These final projects are to be shared in an open house for the wider community.

Examples of children's Tangible K final projects include a robotic city, a zoo with moving animals, a dinosaur park, a circus, and a garden with robotic flowers responsive to different sensors. These projects incorporated the use of inexpensive recyclable materials. For example, one kindergarten classroom in Boston, after a field trip to the old city, constructed a robotic Freedom Trail, using cardboard boxes to recreate the historical buildings of the city and embedding light sensors and motors into the boxes to bring their buildings to life.

To supplement the structured challenges, two to three hours of free exploration are allotted throughout the curriculum. These open-ended sessions are vital for children to fully understand the complex ideas going on with their robotic creations and programs. The free-exploration sessions also serve as a time for teachers to observe students' progress and understandings. These sessions are as important for learning as the lessons themselves! In planning and adjusting the time frame of this curriculum, free-exploration sessions should not be left by the wayside. Rather, if time is tight, teachers can consider leaving out a particular lesson altogether, giving children enough time to really understand and work with the ideas they are introducing rather than skimming over all the lessons presented in this curriculum. Free exploration provides opportunities for playing with materials and ideas. This will help build a solid foundation.

To date, the DevTech group at Tufts University has developed several curriculum units for use with a variety of robotics kits such as LEGO Wedo, LEGO Mindstorms, and KIWI, a developmentally appropriate robotic kit that my team and I designed for early childhood education. The units have been piloted by children in Grades preK–2.

Using the KIWI system, children program their robots by connecting together wooden blocks that are then read by the robot. Each block corresponds to one action that can be performed by the robot. Thus, programming the robot's behavior involves arrangement of a logical sequence of actions represented by wooden blocks. The curricula provide a hands-on introduction to a selection of computer programming and robotics concepts and powerful ideas that are integrated with mathematics, science, social studies, and language arts core curriculum framework.

The curriculum development work described earlier has been described in a number of publications, including Bers (2008, 2010), Bers, Seddighin, and Sullivan (2013), Sullivan, Kazakoff, and Bers (2013), and Kazakoff, Sullivan, and Bers (2013).



Image courtesy of Marina Bers.

Instructional Considerations

Modifications. Some students may benefit from further division of the activities into smaller steps or from more time to explore each new concept before moving onto the next, either in the context of free exploration or with teacher design challenges. Each of the powerful ideas here can easily be expanded into a unit of study. For instance, students could explore a range of different activities and challenges with sensors to learn how they work in more depth.

Teacher as Facilitator. The theory of constructionism developed by Seymour Papert shows that children learn best when they construct digital artifacts and knowledge by playing with and exploring concrete materials. The social context of these explorations is also crucial, and teachers can provide scaffolding by creating a learning environment that supports children's explorations and experimentation. Through questions and observations, the teacher engages students in articulating and extending their own observations, thought processes, and explorations. The teacher may not directly answer students' questions but rather show them how to find it themselves. This kind of exploration fosters an environment in which what we often see as "failure" is a natural step of the learning process, a signal to ask questions and explore further.

The Design Process. The engineering design process of building and the computational thinking involved in programming foster *competence* in computer literacy and technological fluency. The classroom practice of having children keep design journals during the process of creating robots helps make transparent to the children (as well as teachers and parents) their own thinking, their learning trajectories, and the project's evolution over time. Like the scientific method, the formal steps of the engineering design process—posing a problem, doing research, planning, developing a prototype, testing, redesigning, and sharing solutions—give students a tool for systematically addressing a problem.

Journals and Learning Style. Tangible K design journals may provide more or less structured paths for children to navigate the process from idea to product by scaffolding these formal steps. A journal may have worksheets to address all steps of the design process or simply white pages to invite imagination; at best, they have a combination of both. This individualization is important. Some children need constraints and top-down planning in order to work effectively. Others do not like to plan in advance. They might belong to a group of learners characterized as "tinkerers" who engage in dialogues and negotiations with the technology. They enjoy working bottom up, messing around with the materials to come up with ideas as they create, design, build, and program. Both learning styles are conducive for building competence in the technological domain.

Creativity. The Tangible K approach is based on the promotion of creativity, as opposed to efficiency, in problem solving. The approach is informed by the original meaning of the word *engineering*, which derives from the Latin *ingenium* meaning "innate quality, mental power, clever invention." The program integrates media such as LEGO pieces, motors, sensors, recyclable materials, arts and crafts materials, and graphical elements from the programming language. In the process of solving technical problems in creative ways with these media, children develop *confidence* in their learning potential.

Managing Frustration. However, clever or creative projects may be difficult to make, and the process can be frustrating. After many tries, the jaw of a child's robotic crocodile still may not open or her car may break every time it turns to the left. To avoid frustration, some teachers carefully choose the projects for children to work on or provide step-by-step directions. Such a strategy may shelter children from what Alan Kay calls the "hard fun" of creative learning. Instead, the Tangible K approach aims to help children learn to manage frustration—an important step toward the development of confidence in one's ability to learn. The learning environment is set up to create a culture in which it is expected that things may not work and in which succeeding the first time is seen as a rarity, perhaps as a sign that the child might not have challenged herself. As children go through the program, they gradually realize their ability to find solutions by trying multiple times, by using different strategies, or by asking for help.

Collaboration. Most educational robotic programs for older children, such as the National Robotics Challenge and FIRST (For Inspiration and Recognition of Science and Technology), are set up as competitions in which robots have to accomplish a given task, usually with the goal of outperforming other robots. However, research has shown that most females do not respond well to teaching strategies that stress competition; such strategies also might not always be appropriate in the early childhood setting. The Tangible K learning environment, instead of focusing on competition, promotes sharing resources and *caring* about each other.

The use of collaboration webs fosters collaboration. At the beginning of each day of work, each child receives, along with the design journal, a personalized printout with his or her photograph in the center of the page and the photographs and names of all other children in the class arranged in a circle surrounding that central photo (see p. 133). Throughout the day, at the teacher's prompting, each child draws a line from his or her own photo to the photos of the children with whom he or she has collaborated. (Collaboration is defined as getting or giving help with a project, programming together, lending or borrowing materials, or working together on a common task.) At the end of the week, children write or draw "thank-you cards" to the children with whom they have collaborated the most.

Communication. Communication is an important feature of the Tangible K curriculum, which includes mechanisms that promote a sense of *connection* between peers or between peers and adults. One feature that encourages communication is technology circles. During technology circles, children and adults stop their work, put their projects on the table or floor, sit down in a circle together, and share the state of their projects. This is similar to other circle times that children are exposed to in kindergarten.

Technology circles present an opportunity for debugging as a community—that is, for solving technical problems in programming or building. The teacher starts the technology circle by asking children to show their projects and asking questions such as "What worked as expected and what didn't?" "What are you trying to accomplish?" "What do you need to know in order to make it happen?" The teacher then uses children's projects and questions to highlight powerful ideas illustrated by the projects. The curriculum emerges based on what this particular learning community needs to know. This approach provides technical information on demand, based on emerging needs, and is an alternative to lectures. Technology circles can be called as often as every 20 minutes at the beginning of a project or only once at the end of a day of work, depending on the needs of the children and the teacher's need to introduce new concepts.

Community Building. Community-building techniques in Tangible K programs scaffold support networks that promote each child's *contribution* to the learning environment and community. In the spirit of the Reggio Emilia approach (started in municipal infant-toddler centers and preschools of Reggio Emilia, Italy, after World War II), the children's projects are shared with the community via an open house, demonstration day, or exhibition. An open house provides authentic opportunities for children to share and celebrate the processes and tangible products of their learning with others who are invested in their learning, such as family, friends, and community members. These public displays make learning visible to others and to the children themselves.

Choices of Conduct. Tangible K activities provide opportunities for children to experiment with "what if" questions and consider potential consequences of their own choices. Choices of conduct are not only made by children. Teachers also make important decisions that affect what the children do. For example, if the LEGO building pieces are sorted by types and placed in bins in the center of the room (instead of given to each child or group as a presorted robotic kit), children learn to take what they need without depleting the bins of the "most wanted" pieces, such as special sensors or the colorful LEGO mini-figures. They also learn how to negotiate for what they need.

For teachers using the Tangible K program, helping children develop an inner compass to guide their actions in a just and responsible way is as important as the focus on learning about robotics. The program's emphasis on choices of conduct may provoke examination of values and exploration of *character traits*. Differentiation of roles can be important to the growth of a responsible learning community. In any classroom, for example, one child may learn very quickly about mechanics, while another may become a programming expert, and still another may easily problem-solve or skillfully mediate conflicts among group members. Such children may be assigned "expertise badges" by teachers or by the other children. Those children who are seen as especially skilled at something can make the choice to help classmates build a bigger structure or address other challenges. Children are also encouraged to take on new roles and be flexible; there is a badge for "expert on trying new things."

The research that supports the instructional methods described in this section has been documented in a number of publications, including Papert (1980, 1991), Bers (2008), and Lee, Sullivan, and Bers (2013).

Connections to the Next Generation Science Standards

The powerful ideas underlying the Tangible K curriculum were developed to support *Standards for Technological Literacy* (ITEA, 2000, 2005, 2007) and the *Massachusetts Science and Technology/Engineering Curriculum Framework* (Massachusetts Department of Education, 2006). However, the curriculum is also well aligned with all three dimensions of the NGSS (NGSS Lead States, 2013): science and engineering practices, core ideas, and crosscutting concepts.

Science and Engineering Practices. Appendix F in the NGSS describes the eight practices of science and engineering that are appropriate for students in Grades K–2. In our opinion, the Tangible K curriculum provides an environment and series of challenges to meet all eight performance expectations at the K–2 level. That is, students define the problems

they wish to solve (with assistance), develop and use computational models, design solutions, argue from evidence, and so on. However, regarding the practice of mathematics and computational thinking, the Tangible K curriculum enables students to meet the following expectations defined at the middle school level, albeit at a level that is less sophisticated than would be expected of middle school students.

- Create algorithms (a series of ordered steps) to solve a problem
- Use digital tools and/or mathematical concepts and arguments to test and compare proposed solutions to an engineering design problem

Core Ideas. Three core ideas in engineering design are defined in the NGSS for Grades K–2:

K-2-ETS1-1. Ask questions, make observations, and gather information about a situation people want to change to define a simple problem that can be solved through the development of a new or improved object or tool.

K-2-ETS1-2. Develop a simple sketch, drawing, or physical model to illustrate how the shape of an object helps it function as needed to solve a given problem.

K-2-ETS1-3. Analyze data from tests of two objects designed to solve the same problem to compare the strengths and weaknesses of how each performs.

The powerful ideas underlying the Tangible K curriculum are essentially problem-defining and problem-solving tools that enable students to accomplish the above in the context of computational thinking. For example, in order to program a robot, the students need to first define the problem they are attempting to solve. The robots are physical models that are put through their paces to test the students' creative solutions. And the students frequently test their creations and compare their results with those of other students who were attempting to solve the same (or a similar) problem.

Crosscutting Concepts. Appendix G in the NGSS describes seven crosscutting concepts that are appropriate for students in Grades K–2. Here too the Tangible K curriculum helps students develop their understanding of crosscutting concepts. For example, programming computers helps students recognize patterns, especially when using loops and sequences. The crosscutting concept of cause and effect is reinforced as students program their robots to respond autonomously to input from their sensors. The crosscutting concept of systems and system models is also evident in the Tangible K curriculum because the robot is both a system of interacting parts and a model of a system, as it mimics the ability of a person to react autonomously to the input from our senses.

Connections to Other Areas of the Curriculum

The Tangible K robotics program is explicitly designed to address “the missing middle letters” of STEM in early childhood education—the T (technology) and the E (engineering). However, for the program to be successfully adopted in classrooms, it must integrate and facilitate the introduction of other curricular content, both in terms of themes and disciplinary concepts and skills (Bers, Ponte, Juelich, Viera, & Schenker, 2002). Following are several clear connections between powerful ideas from the Tangible K curriculum and other domains of knowledge.

Sequencing, Control, and Flow. A sequence of instructions can be described in a program and acted out in order by a robot. Disciplinary connections: literacy: storytelling, “how-to” books, symbolic system, setting up controlled experiments, basic explorations of geometry, cause and effect.

Loops. Sequences of instructions can be modified to repeat indefinitely or in a controlled way. Disciplinary connections: expanded geometry explorations, time, cycles, symbols used to communicate a message.

Parameters. Some instructions can be qualified with additional information. Disciplinary connections: expanded geometry explorations, number sense, timing and control.

Sensors. Devices measure a change in the environment and convert it into a signal that can be read by an observer, an instrument, or a robot. Disciplinary connections: natural and human-made world; biology: sensing in animals and humans.

Branches. Some instructions in a program ask questions and, depending on the answer, have a robot do one thing or another. Disciplinary connections: cause and effect, logic, expanded scientific observations, executive, function skills, decision making.

Theory

Computational Thinking. The Tangible K curriculum introduces young children to *computational thinking*, which can be defined as a type of analytical thinking that shares many similarities with mathematical thinking (e.g., problem solving), engineering thinking (designing and evaluating processes), and scientific thinking (systematic analysis). The foundation for computational thinking is abstraction—abstracting concepts from cases and evaluating and selecting the “right” abstraction. It relies on selection of inputs (manipulation of variables and computational instructions), observation of outputs (outcome data), and decomposition of what happens in between. Computational thinking involves the ability to abstract from computational instructions (programming languages) to computational behaviors, to identify potential “bugs” or errors to fix, to decide what details among the input-computation-output algorithm to highlight and retain and what details to discard. Wing (2006) describes computational thinking as a fundamental skill for everyone, not just for computer scientists.

The term computational thinking grew out of the pioneer work of Seymour Papert and colleagues on design-based constructionist programming environments; the term was used to refer to ways to algorithmically solve problems and to acquire technological fluency (Papert, 1980, 1991). Previous work on elementary school children and computational thinking can be found in the research literature on constructionist programming environments (Clements, 1999; Flannery & Bers, 2013; Reppenning, Webb, & Ioannidou, 2010; Resnick et al., 2009).

Theoretical Framework. The theoretical foundation that guided the development, implementation, and evaluation of this robotics curriculum is called Positive Technological Development (PTD). The PTD framework is a natural extension of the computer literacy

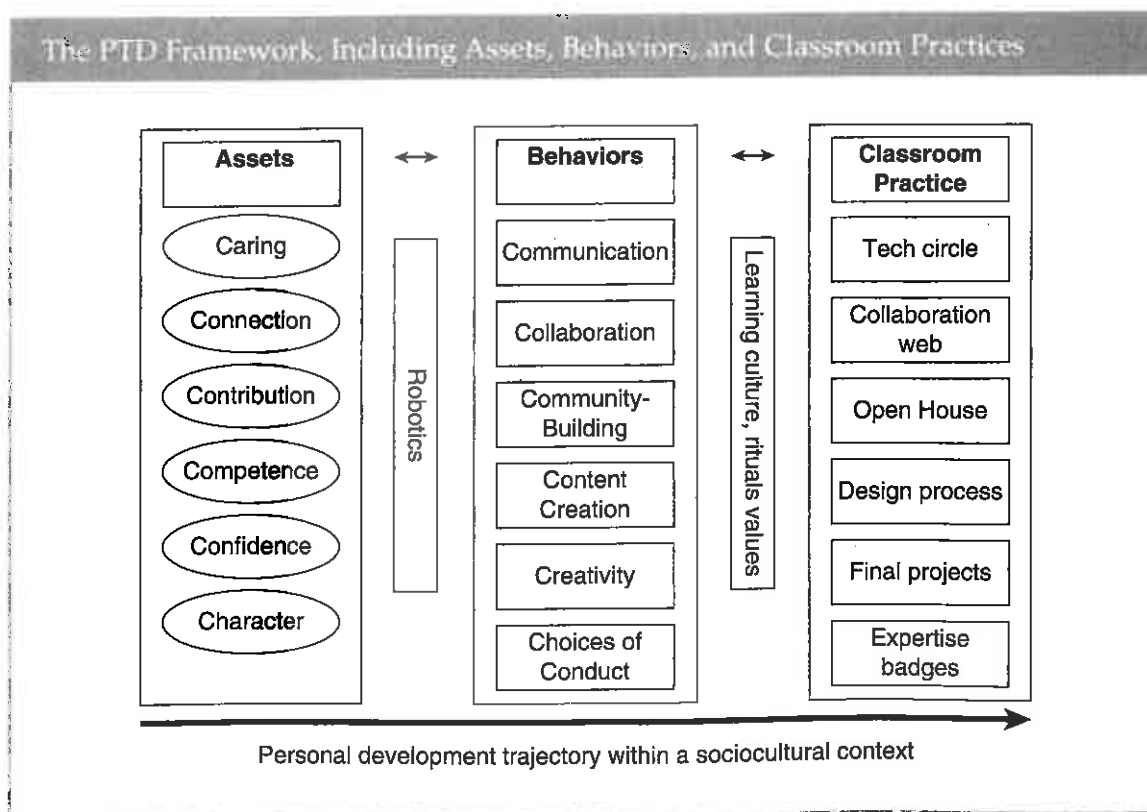
and the technological fluency movements that have influenced the world of education but adds psychosocial and ethical components to the cognitive ones. PTD is an interdisciplinary approach that integrates ideas from the fields of computer-mediated communication, computer-supported collaborative learning, and the Constructionist theory of learning developed by Seymour Papert, and views them in light of research in applied development science and positive youth development. Development of the PTD has been documented by Bers (2010, 2012).

PTD provides a model for developing and evaluating technology-rich programs by focusing on the positive ways that children can interact with technology. The model is illustrated and briefly described below.

Assets of youth development identified by decades of research on positive youth development (Bers, Doyle-Lynch, & Chau, 2012; Lerner, Almerigi, Theokas, & Lerner, 2005) are listed in the left-hand column: caring, connection, contribution, competence, confidence, and character.

Positive behaviors that should be supported by educational programs that use new educational technologies such as robotics are listed in the center column. These include communication, collaboration, community-building, content creation, creativity, and choices of conduct.

Classroom practices that support and encourage positive behaviors and assets of youth development are listed in the right-hand column. These include tech circles, the collaboration web, open house, design processes, final projects, and expertise badges.



PTD takes into consideration the learning environment and the pedagogical practices, cultural values, and rituals that mediate teaching and learning, and therefore the introduction of a robotics curriculum such as Tangible K (Rogoff, Goodman Turkanis, & Bartlett, 2001).

Assessment

The PTD framework shown on the previous page provides guidelines both for designing the educational program and for evaluating children's learning and development. *Content creation* and *creativity* (the first two "Cs") are evaluated in terms of competence (or level of understanding) and confidence in the domain. The following are used to assess content creation and creativity:

- **Student's portfolios** are composed of student's design journals, their programming samples (code), and robotic projects. Change over time in the level of sophistication and complexity is assessed.
- **Video journals** are recordings made at least three times during the program (e.g., beginning, middle, and end) showing what the children have been working on and explaining their activities.
- **A rubric of levels of understanding** is a set of questions for the teacher or researcher to complete at the end of each session to assess each child's level of understanding on a scale of 0 to 5 for each learning objective.

Children's collaboration and communication skills are evaluated in terms of the levels of caring and connection achieved by the children by analyzing the collaboration webs over time (Lee et al., 2013) and the children's participation in the technology circles.

Finally, community building and choices of conduct are evaluated by looking at a child's overall participation and engagement in the Tangible K program and her *contributions* to the learning environment, in particular during the final project presented at the open house. Expertise badges are seen as representative of the child's character traits. Change over time is analyzed.

Assessment in the Tangible K program, unlike most programs focused on technological literacy, addresses not only the cognitive dimension but extends also to the social and the moral dimensions of the child's experience through and with the technology, toward a goal of helping the child develop in an integrated and holistic way.

In terms of research, the evaluation goals guiding the Tangible K robotics program are twofold. The first goal is to provide an evidence-based systematic account of children's learning of components of the PTD framework. The second goal is to establish potential learning trajectories of design tasks with incremental levels of difficulty, matched to the children's levels of understanding (Clements & Sarama, 2009). Future work related to Tangible K robotics will focus on developing new curriculum modules, implementing a less costly robotics system that uses everyday materials, and constructing a solid theoretical model for learning trajectories in this area.

Conclusion

The Tangible K curriculum presented in this chapter brings engineering and computational thinking to the early childhood classroom. It does so by focusing on six powerful

ideas: robotics; the engineering design; sequencing, control, and flow; loops and parameters; sensors; and branches. Following the PTD theoretical framework, this curriculum emphasizes not only the cognitive aspects of learning how to build and program a robot but also the psychosocial and ethical dimensions of teamwork.

References

- Bers, M. U. (2008). *Blocks, robots and computers: Learning about technology in early childhood*. New York, NY: Teacher's College Press.
- Bers, M. U. (2010). The Tangible K robotics program: Applied computational thinking for young children. *Early Childhood Research and Practice, 12*(2).
- Bers, M. U. (2012). *Designing digital experiences for positive youth development: From playpen to playground*. Cary, NC: Oxford University Press.
- Bers, M. U., Doyle-Lynch, A., & Chau, C. (2012). Positive technological development: The multifaceted nature of youth technology use toward improving self and society. In C. C. Ching & B. J. Foley (Eds.), *Constructing the self in a digital world* (pp. 110–136). New York, NY: Cambridge University Press.
- Bers, M. U., Ponte, I., Juelich, K., Viera, A., & Schenker, J. (2002). Teachers as designers: Integrating robotics into early childhood education. *Information Technology in Childhood Education, 123–145*.
- Bers, M. U., Seddighin, S., & Sullivan, A. (2013). Ready for robotics: Bringing together the T and E of STEM in early childhood teacher education. *Journal of Technology and Teacher Education, 21*(3), 355–377.
- Clements, D. (1999). The future of educational computing research: The case of computer programming. *Information Technology in Childhood Education Annual, 147–179*.
- Clements, D., & Sarama, J. (2009). *Learning and teaching early math: The learning trajectories approach*. New York, NY: Routledge.
- Flannery, L. P., & Bers, M. U. (2013). Let's dance the "Robot Hokey-Pokey!": Children's programming approaches and achievement throughout early cognitive development. *Journal of Research on Technology in Education, 46*(1), 81–101.
- International Technology and Engineering Education Association. (2000, 2005, 2007). *Standards for technological literacy: Content for the study of technology*. Reston, VA: Author.
- Kazakoff, E., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal, 41*(4), 245.
- Lee, K., Sullivan, A., & Bers, M. U. (2013). Collaboration by design: Using robotics to foster social interaction in Kindergarten. *Computers in the Schools, 30*(3), 271–281.
- Lerner, R. M., Almerigi, J. B., Theokas, C., & Lerner, J. V. (2005). Positive youth development: A view of the issues. *Journal of Early Adolescence, 25*(1), 10–16.
- Massachusetts Department of Education. (2006). *Massachusetts Science and Technology/Engineering Curriculum Framework*. Boston: Author.
- NGSS Lead States. (2013). *Next Generation Science Standards: For states, by states*. Washington, DC: The National Academies Press.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.
- Papert, S. (1991). What's the big idea: Towards a pedagogy of idea power. *IBM Systems Journal, 39*(3–4), 720–729.
- Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education, 265–269*.
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., . . . Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60.
- Rogoff, B., Goodman Turkanis, C., & Bartlett, L. (2001). *Learning together: Children and adults in a school community*. New York, NY: Oxford University Press.
- Sullivan, A., Kazakoff, E. R., & Bers, M. U. (2013). The wheels on the bot go round and round: Robotics curriculum in pre-kindergarten. *Journal of Information Technology Education: Innovations in Practice, 12*, 203–219.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. Retrieved from <http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf>.

