**David on learning together**

*My Introduction to Computation*

The first time I had the chance to program a computer was in school, although not in a regular class. It was an informal opportunity with an incredible teacher, Mr. Bristol, who always proclaimed mathematics needed to be “true, right, and beautiful,” and who managed to obtain a dial-up connection to a university computer. It is important to note that the primary idea was to “play” with it, not in the sense of playing games on the computer but in playing with a new entity to see what could result without harm.

Programming was new to everyone. It was appealing to some of us because we liked puzzles, challenges, the unknown, and, for me in particular, things that were not compulsory. We did not have lessons. Activity was not pre-planned and pre-ordained. We became peers of a sort with our teacher. He clearly knew more than we did, but we explored together. This relationship and this culture of mutual exploration into the unknown, on problems that we chose to pursue, with colleagues sharing the wonder of the newly possible, was invigorating in ways that, to me, ordinary school was not. Ideas were the currency, not age, authority, or position.

We could think about and work on problems that if we had to approach them by hand, we knew we could not do them (except perhaps with infinite time). While some initial activities were familiar and trivial, sorting a class list in milliseconds for example, the ability to define the mechanism and let the machine do the grunt work was liberating. We were freed to push boundaries. What things that previously appeared impossible, might actually be possible? This sentiment fit with the zeitgeist of the times, challenging ugly elements of life such as racism, unending war and violence, inequality, ecology, thinking that a different world was not just possible but practical.

Thinking in systems, thinking of mechanisms, thinking about more complex thinking about how to describe the world in a functional way and letting machines do the drudge, mechanical work, and then doing such projects was liberating and empowering. Through a now familiar iterative process of writing, testing, debugging, we could refine our thinking until we were satisfied (or sufficiently frustrated to decide to try again another day).

How I now wish we knew about the work of Seymour Papert, Cynthia Solomon, and the Logo group as the 20 things to do with a computer would have been so helpful to point us towards interesting things to do without telling us exactly how to do them, letting us learn, engage, and explore and not just appreciate what they did but to think in better ways, as well as to come up with our new list of 20 things to do.

*Beginning to Program for Real*

Because programming was at the time a rare skill, even as a student I was able to work on real projects, including analyzing satellite data to help determine the state of deforestation and beach erosion in New Jersey; designing and implementing a database and applications to eliminate corruption, enable fairness, and speed the process and thereby lower costs for an urban development; searching to see if there are sequences in non-abelian groups of higher order; and others. In each case, the work could not be done quickly or at all without a computational approach. Paper and pencil could not suffice.

Before commencing on the project in group theory, I knew absolutely nothing about the area. With the help of the professor, by programming the project I learned key concepts in group theory in a way that I personally would not have learned in a typical course. I had to make the ideas functional it in order to explore a real hypothesis in the field. I did it to help discover something new about the world, not to personally discover what everyone else already knew. Moreover, for me, the process of working to make or discover something new in the world provided the best motivation and environment for learning.

There are many genres of computer programming, but designing and developing applications particularly suited me. There is not space here to go into any meaningful detail about the projects, which included an environment for the better practice, research, and administration of healthcare; training air traffic controllers; administering environmental protection; predicting rainfall for flood prevention; and others.

In every case one primary goal was to design and develop *environments for others to* ***do*** *more things better with computers*, and to provide tools and for them to extend, customize, and make their own for what they want to do. In this way I thought of every environment as a learning environment and developed the idea of emergent design in order to design carefully while remaining open for adaptation and evolution as the world changed, as people learned to do more, and to thoughtfully adjust for unknowns and uncertainties.

When I worked in the AI technology center at Digital Equipment Corporation, a Vice-President of Software once said “No one wants to buy a computer. They want to do things.” Doing things to help people do more things is immensely gratifying. Helping to develop new cultures that can take advantage of new things to do in order to help create more just, equitable, and environmentally friendly worlds is immensely more satisfying, particularly when helping to provide environments for populations that have been excluded.

Excluded populations include children. Papert and Solomon saw early that in the same ways computers would come to facilitate the rapid growth of knowledge in the world, which is by definition learning, so too could creative, expressive, constructive, and collaborative uses of computers could also facilitate children’s learning in ways technologies, including pencil and paper, could not.

Solomon wrote that the impact from their work was not just from a particular programming language, but from “ideas, people, a culture, and a language.” All are essential to giving power to doing things with computers. Not having any one this minimum set diminishes potential returns.

Utilizing experience designing and making software and environments for organizations to enable learning, improve collective efficacy, and evolve their practices informed the types of things I would subsequently do with children and communities. My goal was not to try to teach a bit of programming, or mathematics, or electronics/robotics/making, or something else out of the context of use, although learning programming, mathematics, expression, communication, and documentation of ideas was always a major underlying component of activity. The overarching goal was to develop better environments for learning and doing by mindful learning and doing in an integrated fashion on projects of interest. I will highlight one such example.

*A Case Study of Professional Learning*

Air traffic control (ATC) is highly complex and certainly critically essential to get right. Otherwise people die. To their credit, air traffic controllers understand the magnitude and importance of their everyday work. This is the primary reason burnout in the profession is so high. Therefore, when training people to become air traffic controllers, the authorities know the learning must be correct, complete, deep, and real, not an area where someone can pass the exams but not really know what they are doing.

I was trepidatious coming to the national training center for the first time. I was told that almost all the administrators and trainers were military or ex-military and I was dubious that they would be open to the type of learning that would be fundamental to the system I would design. I fully expected to have a quick trip to Oklahoma City, have meetings, a nice meal, and an early flight home.

To my surprise, the administration and the people in charge of learning and training were among the most progressive educators I have ever met. They knew and understood the importance of real learning and the construction of knowledge. They did not merely acknowledge the relevance of Piaget and Vygotsky, they were insistent upon applying their theories. They did not want a rigid, regimented, mind-numbing, drill-and-kill approach any more than I did. They wanted the best in learning and not the worst of traditional classrooms because of the existential importance of truly learning the complex task.

In retrospect, it should not have been surprising. Of course they require the best in learning. Not learning was not an option. Because the criteria for assessment were objective and because the teachers and evaluators themselves were knowledgeable practitioners, they knew if people learned appropriately or not. As the ATC system was in crisis due to increasing air travel, more and faster planes in the sky, the high cost of traditional training and lack of equipment which was based upon the actual machines and conditions, combined with the need to replace the managers and previous retirees brought back into the system after the government fired controllers in an effort to bust their union, the need to properly train people was critical. People were retiring faster than replacements could be trained.[[1]](#footnote-1)

Creating computer-based simulations was one obvious and essential step. However, learning by working with simulations in an unguided manner where others determined the rules of the simulation that were opaque to the learner was insufficient. Thus, there was a need to put intelligence in the system. What evolved, though, was a system that used its intelligence to help people to better use theirs.

The computational complexity of the domain, with so many unknowns, uncertainties, and possibilities, made it such that it could not be completely automated. The expert air traffic controllers told us that there is no one right way to perform properly. People needed to develop their own styles and capabilities. People needed to make and learn from mistakes during training in order to better understand the problem space and find their best methods to do their work. They knew that to become a good controller was not to memoize and follow rules, to passively try to apply rules without regard to circumstance, but to deeply learn, to assimilate the best ideas, and to accommodate oneself to adapt to how one needs to perform better.

However, because ATC trainees are paid while learning, and can be dismissed if not deemed up to the task, many trainees would not be adventurous during supervised training, fearing to make a mistake and be marked down and potentially dismissed. Because working with real equipment, including simulators, was expensive, they needed a place to play, to try, to experiment. And if that space could also provide intelligent feedback, all the better.

I designed a system that borrowed from ideas of Marvin Minsky about a “society” of mind, with multiple agents with specific but limited expertise offering themselves to help. However, the trainee remained in control and was the integrator, as this was an essential part of what they needed to learn. I also borrowed from interface ideas of Muriel Cooper where the agents presented themselves more strongly or weakly based upon system knowledge and prior learner choices. However, the path through the material remained with the learner so they could focus on what they needed at the moment. Using analytics the system could demonstrate the certain essential ideas and practices that needed improvement or that were being neglected. Trainees would work through the control scenarios, availing themselves of agents as they thought best, try out their crazy ideas, work through scenarios, and, with computational and personal assistance,

People, experienced and novice, adored the prototype system. However, again what in retrospect should have been obvious but what attracted attention at the time, was that what they all liked best was the scenario generation tool I made for myself and team to expedite making training scenarios.

They valued working through the intelligent simulations and then debugging performance with the experts. However, the group preferred working together to define the scenarios that would raise the key and difficult ideas in ATC. The tool was defined symbolically, so to define scenarios one would name the types of problems and sub-problems to be encountered, in sufficient detail for the system to develop scenarios where the conditions arose. They believed that collaboratively, experts and novices together, discussing what initial conditions might lead to which problematic conditions and then require certain knowledge and decision-making to resolve a potentially dangerous situation, was the best of all the learning experiences. They could discuss, program, and then run their scenario, pause it as needed, discuss some more, try different things so that they could see what worked and why it worked (or not), re-run it as often as they liked, tweak some parameters, try again, and also use the feedback from the system as another voice (which also helped debug and improve the system). They preferred a microworld for ATC.

Each air traffic controller in the field controls a 3-dimensional space, and tells the aircraft pilots what to do. Storms, strong winds, flight delays, and other factors create uncertainty and with so many planes in the air, each with their own goals and constraints (for example, unlike a train, you cannot tell a plane in the air to just stop where they are and wait), the complexity can be overwhelming.

One of the biggest problems is when a controller of one space needs to “hand-off” control to the controller of the proximate space. Problems arise when controllers solve their particular problem but then complicate the person receiving the aircraft into their space. The scenario generation tool and multiple environments on different computers for different trainees provided a means to think about, design, and test the situations so these complications would be less likely to arise. Training, which is so often individual, could be collaborative and problems in coordination could be addressed. In this way they could develop individual and collective efficacy.

The training had some advantages over typical school environments. The ratio of teachers to learners was low. The teachers were learning and doing with the students, and not just transmitting information about things. There were people with a mixed range of experience working together, as controllers would return from the field for re-training. They had the advantage of sufficient numbers of computers for all people. Spending was appropriate to enable all to learn. They were learning by doing and reflecting upon what they do. They created an environment that encouraged big thinking, trying out ideas, including crazy ones. They created an environment where each supported the other and was not competing with them.

Educational systems do not make such choices. However, they could. There are demonstrably better ways to do and organize learning environments. There does not have to be a world where more is spent on weaponry that on children’s education. These are questions of our priorities. Thankfully, people in charge of air traffic control education make choices to improve the safety and functioning of their part of the world. It would be better if we made the same types of choices for and with our children.

*Some Principles for Learning Environments*

Papert and Edith Ackermann emphasized that our preferred activities are “hands-on and heads-in,” to differentiate the activities from students being told information about things but not engage in doing them concretely, and from mindless making, where one might make things but not necessarily learn or engage in the what and how of making.

Mel King described our collaborative activities as designing and making things supported by caring others while building community and providing opportunities to those typically excluded.

Paulo Freire inspired our work to engage people in conversation with their community. Freire concerned himself with developing critical consciousness, where literacy was an empowering tool. To us, computational literacy in the deepest sense of literacy was another empowering tool. Freire emphasize the importance of generative themes were all participants would have their own ideas of what to do within the theme.

We adopted, integrated, and complemented these fundamentals in our work with children. We created cultures of creativity and learning focused on learning, but in the context of making a better world, of social justice, of developing community by making things for one’s community. Just as in my early experiences programming, we all learned together through exploration and investigation. Ideas and expertise were valued, not positions. We all learned and taught as partners. We all researched understanding of human learning through our projects, most often in pursuit of improving community, justice, and positive development.

Again, space will not permit a full, detailed accounting of all or any of the projects. We worked in schools and communities across the world, including a project in Thailand to help modernize the educational system and where several villages transformed the socio-economic conditions through a constructionist learning approach using computation to change local conditions; creating an alternative school in the Maine juvenile jail where the youth involved transformed their lives through a different approach to learning with computation; creating a variety of programs for youth at the South End Technology Center of Boston focused on designing and building with computational technologies to improve life in their communities, and more. While all of these projects are in some ways different, for different people in different places, in other ways they are the same by *providing better things to do with computers with others and in learning environments designed to facilitate learning, doing, sharing, and community*.

In each case, while the projects differed and built upon each other, they shared the same underlying principles of learning through programming and making artefacts and models of what the learners deemed important, particularly in order to improve their communities. By doing these projects in the environment we jointly created, they learned how to learn by making and reflecting on their thinking, to problem solve by working through the myriad problems encountered in actualizing their projects, to think critically in order to determine what to do and how it might help community, to develop individual and collective efficacy by making their projects and positively critiquing projects of their colleagues. The environment was integrated and whole. They were not just to follow directions but to imagine, think, create, and debug. We worked together, thereby working in a more egalitarian, democratic way, and thereby learning about democracy and society by doing, not by being told information about the concepts.

In Maine, Boston, Thailand, Senegal, Rwanda, Costa Rica, Brasil, Colombia, and elsewhere, we worked on projects students chose to improve their communities. Many projects were brilliant and, at times, actually put into practice in the real world, not just as school models. I choose just one to demonstrate a way of working.

*A Case Study*

In Brasil we developed a project called “The City We Want,” supported by the public schools of the municipality of São Paulo, Rodrigo Mesquita, and by the Bradesco Foundation. In a residential school in Bodaquena, in the Pantanal, the world’s largest inland wetlands and an unfortunately threatened ecosystem. An important project for the students was to research how to cool their classrooms and dormitories. The climate is intense. On the day we arrived the temperature was 40C (104F) with humidity of 90%. They had some fans, but not enough for every student. The rooms had 2 windows on each of 2 walls, and a small window at the second level over the door.

How could they cool the room? They assumed that all of us, coming from MIT and having PhDs or being graduate students would just know the answer. Perhaps we should have, but none of us did. The students looked at us like we were frauds. How could we not know? However, when they saw how we were excited about figuring this out, and did not try to hide that we did not know, they began to see intelligence as not already knowing an answer to something, but as the ability to figure something out correctly.

We began to discuss and decide how to solve this problem. The world talks of the importance of learning to learn, particularly now in a rapidly changing world, but exactly how to learn to learn, and not just be told, is under-explored and under-theorized. The teacher is the most experienced learner in the classroom, but schools are engineered to never let the students observe the teacher learn anything. The process of working together to collectively solve the chosen problem, with our greater experience in research, problem solving, and using the technologies at hand was probably the most meaningful aspect of the experience.

The students determined how to set up the experiment, which, naturally, was not correct on the first pass. They only checked the indoor temperature, since they figured that was the primary determining factor for comfort. We pointed out that if they did not know the outside conditions, its temperature, whether it was sunny or cloudy, how windy it was, and perhaps other factors, they would not know enough to arrive at a real solution.

We worked on what were the factors that led to comfort or discomfort. They had to think about why a fan cooled people. They knew comfort was not merely due to temperature and also involved humidity. But what exactly is humidity and how do you measure it?

As we explored there were many ideas about what to do and how to arrange the fans for maximum comfort for all. Cross ventilation was not possible. Heat rises. Should they point the fans upward to push the air out the top window and cooler air could perhaps enter from the first floor windows? Should they try to blow the air out the windows? Should they try to arrange the fans in a circular pattern?

As they experimented they also began to look at other factors. What was the composition of the roof? Could they make an automatic sprinkler system to cool the roof and would that have more impact? If they placed buckets of water in front of the fans, would that help?

The technologies facilitated running multiple experiments, data collection, graphing, comparisons, and successive refining of ideas. In pursuit of a solution, the underlying mathematics and physics was necessary. They were not studying issues such as heat transfer, conduction, air flow, and the mathematics of measurement in the abstract, to be forgotten after the exam. They needed the ideas to solve their chosen problem.

They eventually arrived at a solution, but even with their graphs, they felt dissatisfied about understanding what was really happening. Because it was a residential school and had a small farm where they kept bees, they used the bee equipment to generate smoke and videotaped its dispersal and flow as the fans operated. This further made concrete what was happening. As they told us later, the downside was that the smoke remained in the classroom for 2 weeks, which was another illuminating lesson. Perhaps the most important lesson, was that they had the agency and capability to work on and solve difficult problems.

*Some More Principles*

Whether cooling a room in a tropical climate, determining the health of the forest in Thailand, water management almost everywhere, improving quality of life in an arid city in Chile, reducing violence in the periphery of cities, in each case we did not know the best answer beforehand. However, through exploration, investigation, design, prototyping, testing, analyzing, debugging, and re-formulating, collectively we arrive at a satisfactory place with satisfactory technologies and an improved environment. Moreover, we all learned, in context, and with values, and we learned how we learned. We discussed our learning explicitly, taking the same approach to learning about learning, that we did to learning to Coll the classroom. We struggled at times as these are complex problems, but by overcoming it we developed the socio-emotional factors people now recognize as important. We developed them not by being told about them, but by succeeding at hard tasks.

Our way of working was similar to our way of working with adult and organizational clients. We respected them, their thinking, and their values. We would not try to impose a solution on them. Rather, we worked to co-develop solutions. While working we would strive to make connections to the underlying powerful ideas and knowledge/ Each participant has particular expertise, and collectively we work towards an agreed upon outcome. Often there are surprises, felicitous and less so, along the way, which can be built upon or negated or worked around.

Our way of working was not to just show kids what to do or do things. We engaged in their explorations and projects whether we knew what to do or not. Thinking about how to improve one’s community, for example, is an essential project. Imagining and creating projects to accomplish this is not only a wonderful learning opportunity, it also can be transformative for the person and even the community. Many of the young people are treated as one of the biggest problems in the community. Beginning to see themselves as powerful agents of change is an incredible and transformative outcome.

We needed to understand how they were thinking and making meaning in order to see better how to interact with them to successfully make their projects and acquire the deep ideas. By making concrete artefacts, whether expressed in code or physical materials, gave concrete insight into what they thought. Through debugging it became possible to think about one’s thinking about what they were doing.

I developed the concept of emergent design while working in industry to address the unpredictability of the future, to enable systems to evolve based upon the learning of the participants, and to provide more dimensions for more just, inclusive, and democratic decision making. While working at the micro level with learners as well as working at the macro level to try to facilitate systemic change, emergent design also provided a strong framework to enable more democratic control and better, more just outcomes. Doing things with computers in order to learn in more effective and organic ways functions better when activities are more emergent based upon the people and the situation. We do better and learn better when we can think about and do not in fixed, pre-determined, and externally controlled ways, but when we all can be the protagonists of our own learning, informed by our environments, cultures, and knowledgeable and caring others.

*Conclusion*

The role of the computer as constructive, creative, and expressive material is essential. Without computational materials most of these projects could not be realized. By using the computer only as a delivery mechanism of text and image, we could gain more information about the issues, but by designing, and defining in a formal language on the computer, where that is a concrete view into our thinking and thereby debuggable and available for reflection on our thinking,

When simplified critiques of what technology cannot do, or how computers in classrooms only look at what classrooms have and not what people **do**, in classrooms, we are left with an inaccurate and misleading picture. With the current pandemic, we witnessed that a vaccine could be developed within ONE WEEK from the discovery of the structure of the virus. Neither the vaccine nor the structure of the virus could be discovered so quickly without the computational technologies available to the scientists. Fortunately, in medicine and public health people do not try to reduce the activity to having technology or not. People do not say that the technology by itself solved nothing. Why would we expect otherwise, either for vaccine development or for learning?

The 20 Things article pointed us towards a recursive list of things to do for learning. However, there are also recursive lists for the other elements in Solomon’s formulation. There are many possible cultures, many powerful ideas, and many people with whom to do these things, without a unique one right way to do them.

I describe my trajectory briefly here. Others describe theirs. It is not that only one is the right one, or even that all learners would resonate with the same one. What does matter is to create and evolve better learning environments for all, that bring the possibility of many wonderful, engaging things to do that provide access to powerful ideas and deep learning, many ways to do them, in many different settings, with a diverse set of knowledgeable, caring people, utilizing the best principles for learning, and discovering more along the way, actively and collaboratively constructing and re-constructing the environment and the knowledge.

1. Learning about the crisis, and especially after a Friday bull session over beer where the expert controllers recounted stories of near disasters made me consider walking back to Boston. [↑](#footnote-ref-1)