Working in ToonTalk with 4- and 5-year olds

Leonel Morgado, Maria Gabriel Bulas Cruz, Ken Kahn

USING ToonTalk™ IN KINDERGARTENS

Leonel Morgado, Maria Gabriel Bulas Cruz
UTAD
Eng. II, Quinta de Prados, 5000-199 Vila Real, Portugal
leonelm@utad.pt

Ken Kahn
Animated Programs
49 Fay Avenue, San Carlos, CA 94070, USA
KenKahn@ToonTalk.com

ABSTRACT

We aim to find ways, tools and/or techniques that allow children in this age span to use programming language concepts to express rules and play with them. ToonTalk, with its visual, animated environment, and program-by-example methodology, seemed a very nice tool to start with.
On this paper, we will present our experience using Portuguese ToonTalk with 7 children (5 were 4-year olds, 2 were 5-year olds), in Vila Real (Portugal) kindergartens, from May to June of 2000.
The most encouraging results, including the children's enthusiasm and the robots they managed to program, will be presented, along with some of the hurdles that require longer, broader research.

KEYWORDS

ToonTalk, concurrent programming, children programming, programming by demonstration, kindergartens

1. INTRODUCTION

Computers have a fundamental presence on our everyday lives, not only as automated elements, reacting to sensors and situations, but as tools that the important ability of being adaptable to different tasks, suiting the needs of their users. This adaptation, by which I refer to the common term “programming”, allows the same tool to do tasks with as little in common as word processing, number crunching, entertainment, etc.

Most users, however, simply use “applications”, programs made by other people. While many applications allow for customisation by the end users, these are skills that elude most of them.

This amounts to an immense wasted potential. Among the most obvious motives, we may quote David Canfield Smith and Allen Cypher (1999), referring to the gap between human communication and computer programming languages as the Grand Canyon. Norman (1986), quoted by Smith & Cypher, proposed two ways of closing this gap: moving the users closer to the system (teaching traditional programming to humans – the traditional way to do things) or moving the system closer to the user (making computer programming resemble human communication). Smith & Cypher recall that traditional computer programming education is hardly a discipline for every user: it requires effort, dedication and persistence (just like learning a foreign tongue). And conclude that computer programming should seek out ways of getting closer to human communication, making it possible for every human being to benefit from the computer tool’s ability to be adjusted to different needs. This approach is mentioned in an ACM paper (Myers 1996), where end-user programming is one of the strategic research and development areas pointed.

Thus, in what ways could this skill be introduced in pre-school contexts? Modern pre-school education, rather than focusing on a curriculum, aims to give children basic skills that allow them to more easily and fully develop their personal worldview and self-learning ability. It seems to us perfect sense to give children the possibility to discover and develop programming skills, centred on learning how to state rules, how others (a computer, for instance) would follow them, and how different rules would interact with each other.

2. PRELIMINARY ISSUES

The preschoolers' limited ability to conceptualise rules and abstractions may be considered, from a broader perspective, to be the main issue at stake. However, perhaps the most obvious hurdle regarding the introduction of computer programming concepts in preschool is the absence of basic reading and writing skills. Although this is a purely technical computing hurdle, and icons or logography-based writing can be used as means of expression for the "programmer's" intentions, it seemed to us that in order to better reach to some conclusions we required a way to allow children to better express their desires.

Using drawing boards as a way to design programs (having a computer educator translate children's drawings into programming, i.e., being an "human compiler"), resorting to physical, theatre-like plays or developing graphical programming tools, suited to our research needs, all were possible starting points.

Luckily however, our initial bibliography and software review showed that we were not alone in this desire to bridge the gap between children and computers. And albeit works dealing with preschool children and programming are fairly limited (ex.: Perlman 1976), some of the methods and tools being used with older children seemed suitable enough for the initial phases of our research. Two programming tools in particular drew our attention: StageCast Creator (previously called Cocoa), and ToonTalk. Among the papers describing these systems, we may point out the ones by Smith & Cypher (1999) and Kahn (1996).

3. THE ToonTalk™ OPTION

ToonTalk is an animated programming language: the code is "animated" as in an animated cartoon, hence the name; in computer science terms, it is an implementation of a concurrent constraint programming language (Saraswat, 1993). In ToonTalk, children control an avatar moving in a city (metaphor for the entire computation). The city has houses (agents, actors, processes or objects), where robots (methods) can be programmed. The programming employs objects such as boxes (tuples, arrays, vectors or messages), scales (comparison tests), trucks (agent spawning), bombs (agent termination), notebooks (program storage), text and number pads (constants), and birds and nests (channel communication). Robots are programmed moving them with the mouse and performing the required actions (method actions), inside their thought bubbles. After training a robot the method preconditions are visible as the thought bubble of the robot.

From the very start, ToonTalk seemed to us to be the most adequate tool for the target age group (3-5): -

- it employs larger control elements, being more easily controlled by children which are still developing mouse-control skills;
- it seemed to be more easily configurable (by developing objects and behaviours within ToonTalk itself) than StageCast Creator, which could allow us to program situations and elements as required, therefore lessening the risk of having to change the programming environment to suit the research requirements;
- it requires the use of only three mouse skills: moving, clicking and dragging, against the more full array required by StageCast, which also employs click and hold, and right clicking;
- several of its functions could be reached by pressing single keyboard keys, which could allow us, if required, to employ keyboard overlays or conceptual keyboards.

In our view, StageCast Creator's main strength is the way it allows a child to devise a story and specify simple behaviour rules in a simple fashion. However, the need to devise rules for all circumstances that might arouse, and the overall look of the application (which, albeit simple, we feared might prove too complicated for widespread usage by preschool children), also contributed for our option to use ToonTalk.

Finally, ToonTalk and StageCast Creator aren't completely language independent: ToonTalk, for instance employs a talking "Martian" as a help system; also, some behaviours of its controls are language-dependent (the button that makes the bike pump enlarge or shrink objects, for instance, has the letter B –big – or S – small – on it). And several elements on the StageCast Creator environment are also language-dependent. Having these language-dependent elements in English would be yet another hurdle for Portuguese-speaking children. This also tipped the scales towards ToonTalk, whose European Portuguese version was launched in May 2000.

THE INITIAL SESSION DURATION ASSUMPTION

In May 2000 (release of the European Portuguese ToonTalk), three Vila Real (Portugal) kindergartens wished to cooperate in our research: As Árvores (AA), S. Pedro Parque (SPP) and Araucária (AR). In each, two children were selected by the kindergarten teachers, for weekly ToonTalk sessions, until the end of June.

Based on our experience with computer activities for children of this age, we opted for 10 to 20 min. sessions. Usually children start to get fed up, when computing activities take longer. We intended to try out this assumption by having 10-min. sessions in SPP, 15-min. sessions in AR and 20-min. sessions in AÁ.

However, from the very first session, this assumption proved to be wrong: children loved the ToonTalk environment and didn't feel bored at all! In fact, we felt more time was required for adequately exploring the children's ideas and let them practice ToonTalk skills. Table 1 details the evolution of time occupation.

Session	Date	Start time	End time	Duration
S. PEDRO PARQUE				**Average duration: 19 min.**
1	May 2nd, 2000	11h42	12h03	19 min.
2	May 23rd, 2000	11h37	11h58	21 min.
3	June 5th, 2000	11h43	12h00	17 min.
4	June 8th, 2000	14h23	14h45	23 min.
5	June 12th, 2000	11h50	12h00	10 min.
6	June 15th, 2000	11h35	12h00	25 min.
ARAUCÁRIA				**Average duration: 34 min.**
1	May 30th, 2000	11h00	11h17	17 min.
2	June 15th, 2000	10h30	11h15	45 min.
3	June 20th, 2000	10h45	11h25	40 min.
AS ÁRVORES				**Average duration: 28 min.**
1	June 9th, 2000	15h20	15h50	30 min.
2	June 16th, 2000	10h45	11h20	35 min.
3	June 19th, 2000	10h30	10h50	20 min.

Table 1 – Time occupation along the several sessions

As we can see, our 10, 15 and 20-minute groups fell, almost immediately, into 20, 30 and 35 minute groups! And the 40-minute and longer sessions made us realize that actual 45 minute to 1 hour sessions were most likely viable. (This was, in fact, the base for the sessions during the following year's research.) It is worth mentioning that session 5 in S. Pedro Parque was shorter than usual due to hardware problems and session 3 in As Árvores was abbreviated, because it was end of year play rehearsal. Counting out these sessions, the average duration rises to 21 and 33 minutes.

RESEARCH FOCUS

Given the small number of sessions that could be performed before summer holidays ensued, we decided to focus them on two main issues: -

- ability to use ToonTalk's method for generalization, given children's limited abstraction skills;
- results yielded by a directed approach, against a coach-style approach.

The first issue was key for further experiments with ToonTalk, since the children's inability to use generalization would severely compromise any options of rule programming for generic circumstances.

The second issue was aimed at how we should proceed with the sessions on the following year: directed (i.e., proposing activities and conducting children on how they might achieve them) or coached (letting children choose what they want to do and help them achieve them).

Two different approaches on the generalization concept were used: one on S. Pedro Parque, another one on Araucária. The directed approach was used on S. Pedro Parque and Araucária, while the coached approach was used in As Árvores.

4. ACTIVITY FOR DIRECTED SESSIONS: THE EXCHANGER ROBOT

Directed sessions were based upon a simple activity: programming an “image-swapping” robot, i.e., one that takes two images and then exchanges their places.

This can be achieved by placing two boxes in the ground, so that they are combined into a two-hole box; different images are then placed on each hole (in our sessions, we used a tree and a flower). Upon giving this box to a robot, we float into its thought bubble, were we can command the robot with the mouse. The robot then is made to pick the flower and drop it outside the box. Then, it is made to pick up the tree and drop in the hole previously occupied by the flower. We conclude the robot’s programming by making it pick up the flower and place it in the hole previously occupied by the flower.

This robot can then be generalized by setting ToonTalk’s vacuum cleaner tool to “Clean” status, under which it “erases” the surface of images on its thought bubble, leaving only a generic, blank picture. However, setting the cleaner to “Vacuum” status, we can vacuum the entire images from the thought bubble, and this also generalizes the robot (the only difference being that by cleaning images the robot still requires that images are provided; vacuuming them makes the robot accept any object).

We used the vacuuming approach, since it is visually simpler, while still allowing us to evaluate the generalization issue.

5. THE CHILDREN

In order to ensure the children’s privacy, we will refer to them by their first name initials (table 2).

KINDERGARTEN	CHILDREN	AGES
S. Pedro Parque	Z (boy) and O (boy)	5 and 4
Araucária	M1 (boy) and R (boy)	4 and 5
As Árvores	J (boy), M2 (girl) and S (girl)	4, 4 and 5

Table 2 – Children identification and ages

6. INITIATION – DIRECTED SESSIONS

We went over the initial ToonTalk environment: identifying the helicopter as such, and training its controls; controlling the hand with the mouse and noticing that it made the object under its pointing finger shake; identifying robots as such and learning that the cloud near the robot’s head was a thought bubble, which contained the robot’s thoughts. Instead of exploring the tools (vacuum cleaner, bike pump and magic wand), the children were led straight into robot programming, learning skills along the way: box manipulation was practiced, but used afterwards to provide parameters to a robot, for instance. Parallel activities (getting pictures from the notepad, shrinking the tree), were conducted in front of them, and sometime by them, but not always part of the children’s activities. The robot programming for performing picture swapping was demonstrated on the first session, and on following sessions replicated by the children.

This is the full list of actions done entirely by the children, while replicating the first session’s demonstration (only action they didn’t execute: picking up the book with images from within the larger one):

- Pick up and drop a robot, then pick up and drop a box.
- Pick up another box and drop it over the border of the first one.
- Seek a tree in the images book, and pick it up.
- Set the pump on the "P" for "pequeno" (“small”) and shrink the tree.
- Place the tree in a hole, in the two-hole box.
- Seek a flower in the images book, pick it up and place it in the box.
- Hand the box to the robot.
- Inside the robot’s thoughts, perform the image swapping.

7. GENERALIZATION – APPROACH 1: EXPLANATION

In S. Pedro Parque, after programming the exchanger robot, the children tried it out, to see that it would only work with the tree and flower in the right positions. Then we explained to them that cleaning the images from the thought bubble would prevent the robot from being so fussy, and demonstrated the method.

This approach yielded no result whatsoever: the children only got confused. Therefore, we simply played around with some ToonTalk elements (trucks, bombs), to avoid pushing the concept through.

On the following session, a theatre-play approach was used for presenting the explanation, using the following physical material as stage props: two square baskets, similar in colour and size to the ToonTalk boxes; an A4/Letter sheet of paper on which we drew a thought bubble with a ToonTalk box (figure 1); two paper squares, one with a tree, another with a flower, attached to the thought bubble's box with scotch tape; two empty A5 sheets of paper (half of a A4 sheet); six markers: 2 black, 2 yellow and 2 green.

Figure 1: the A4 sheet for the robot's thoughts

Figure 2: the A5 sheets for the blue baskets.

The robot's generic behaviour was shown to the children again, which were completely puzzled over it. We then initiated the "robot" play. The children used the markers to draw a flower on the A5 sheets (figure 2), while we drew a tree. One child would play the robot, holding the A4 sheet over his head; the other would fill the baskets with pictures and hand them to the "robot". This way, the child playing the robot would have to check his "thoughts" content before proceeding, which we hoped would help clarify the robot's behaviour.

Only one issue was detected: the child presenting the baskets would have a different perspective of left/right than the robot-playing child! We overcome this problem by acting as "in-between" traders, which would turn the baskets; this would ensure they both had the same left/right perspective.

All worked fine: they would check the though bubble sheet, and the robot would only exchange the pictures when they matched. However, when we pulled off the taped pictures from the thought-bubble sheet, leaving only the drawing of an empty box, a curious behaviour ensued: the robot-player started to exchange the images continuously, without looking at the thought bubble to check their applicability! He was mimicking the apparent robot behaviour. Clearly, the concept of "entry parameters" or conditions, while clear for concrete examples, was completely disregarded for the abstract case of generalization.

A future experiment about generalization could be done in the following way: let us suppose that the flower-tree exchange was to occur within a 3-hole box like [flower | tree | truck] and the generalized version after vacuuming was [| | truck]. Will the children check the thought bubble while playing the robot game (since they will have to check that a truck is still there even though this robot doesn't use it for anything)?

8. GENERALIZATION – APPROACH 2: USEFULNESS

At Araucária, whose sessions started after we had completed two at S. Pedro Parque, we wanted to use a different approach. After a child expressed his wish to make a robot exchange a truck with a bomb, we decided to try and have children program robots to exchange flowers and trees, and robots to exchange trucks and bombs, leveraging generalization as a time- and effort-saving technique.

The programming of robots with distinct entry parameters allowed children to more easily understand the robot-programming concept, but also allowed us to present the case of both robots being, in fact, doing the same thing, except being too picky. The vacuum method of generalization was then presented simply as a time- and effort-saver, allowing us to avoid the pickiness of the robots. We expected this to yield more success than the approach at SPP, but the success surprised us: children saw this as obvious, as a perfectly natural way to do things. Even though the experiment ground was extremely limited, due to time constraints, this nice result, with two children (a 4-year old and a 5-year old), raised our expectations regarding the possibility of larger-scale use of ToonTalk for definition of rules involving generalization.

9. INITIATION AND DEVELOPMENTS – COACHED SESSIONS

In the coached sessions, the children could decide what they wanted to do, and only sporadic suggestions were presented. A much larger focus on the interface tools occurred (the most visible elements in ToonTalk). Birds and nests were also prime attention targets, since they provide a highly animated, amusing activity.

After the initial session, which allowed them to explore these basic activities, we introduced the concept of robot programming. Their choices of programs were more or less connected with the tool manipulation activities: J wanted to make a robot to clean up the room (we directed him to program a box-cleaning robot, specialized on boxes with nests) and M2 wanted to make a tree-chopping robot! Although we could simulate this quite easily, taking advantage of small pictures to crop a larger tree picture, J had unintentionally limited access to the picture book, so M2 ended up using our unoriginal improvised suggestion of an exchanger robot. S decided that she also wanted to make a box-cleaning robot, only specialized on cleaning up bombs from boxes.

These two sessions were the only coached session we managed to fully conduct, because the last one fell on rehearsal day for the end-of-year party. However, on this last, very short session, we still discovered that S, while not being too active in the previous sessions (J and M2 more or less took over control of the mouse), enjoyed playing around with ToonTalk to the point of not leaving it to eat cakes and refreshments along with the other two children: instead, she was creating birds, moving objects around, enlarging and reducing them… Playing, plainly.

The main advantage of this coached approach, it seemed, was the greater control over ToonTalk's objects and tools, and an overall greater ease of operation. However, robot-programming experiments were very simple, and therefore inconclusive. It was encouraging for us, however, that even a quiet child like S would prefer playing with ToonTalk over cakes and refreshments, and that two more active children like J and M2 would be active while at the same time focused on ToonTalk. This coached approach seemed promising enough for further, more lengthy experiments to be conducted over the following year.

10. FURTHER DEVELOPMENTS – DIRECTED SESSIONS

On both kindergartens where directed sessions were conducted, the robot programming and generalization trials were done on the initial two or three sessions. Therefore, trying to collect further information on children's reactions to the several ToonTalk elements, we decided to introduce birds and nests, to check out the children's response. Since these children had already been introduced to robot programming, we combined bird use with robots, to achieve more dynamic (and hopefully enticing) results.

As expected, birds were a success: children loved playing with them, given the rapid response and animation.

On Araucária, where a single robot-and-bird session was conducted, M programmed a robot that would send boxes of robots, endlessly, into a bird's nest. And, most strikingly, had the intuition that by copying a nest, its bird could carry things to both copies. This was a surprising and most encouraging result.

On S. Pedro Parque, we had the opportunity to conduct three sessions with birds. On the first one, we simply went over bird-carrying: using birds to clean up the room into their nests, for instance, and also trying to give the birds different objects. For the following two sessions, only Z was available, and this allowed for more elaborate activities.

On the second session (only 10 minutes long, due to hardware problems), we introduced Z to the concept of working with nests in a robot's boxes (entry parameters). On the third session, we told Z that he could use an exchanger-like robot and combine it with birds. So he programmed a robot that would pick a flower from a box and give it to a bird in another box. This allowed us to suggest to Z that he could place the bird's nest on the box with the flower on top. And Z quickly realized, which we confirmed by questioning him, that this would allow the robot to keep juggling the flower endlessly.

Finally we suggested that he could create a second robot, which he did (copying it with the magic wand) and cross the parameters between them – therefore creating the effect of two robots sending a flower to each other. While we had to perform this last, more complicated procedure, he was following it with keen interest – not like the puzzling situation of generalization at all. This final, briefly appreciated result also encouraged us to pursue ToonTalk activities with young children, with some expectations regarding the degree of complexity they may achieve on a larger number of ToonTalk sessions, spread throughout the year.

11. CONCLUSIONS

ToonTalk is quite a success with children from the entertainment point of view; they enjoy playing with it, even if only by using the provided manipulation tools. The existent generalization techniques seem viable, if presented in a usefulness context; and we have collected nice indications regarding the amount of rule complexity it allows children to grasp, which prompted us to follow up with further, lengthier sessions and research in the following year.

ACKNOWLEDGEMENT

We wish to thank Mr. Secundino Correia at CnotInfor for his support, including offering us a copy of the Portuguese edition of ToonTalk as soon as it became available, and also to the teachers at the kindergartens involved, that offered us all the help we requested and more. Finally, we wish to thank all the children that took part in this research: their joy and enthusiasm while using ToonTalk soon became one of our greatest motivations to pursue this research.

REFERENCES

Kahn, K. 1996. ToonTalk™ – An animated programming environment for children. *Journal of Visual Languages and Computing*. (An abbreviated version appeared in *Proceedings of the National Educational Computing Conference*. Baltimore, MD, USA, 7 (June): 197-217, 1995.)

Myers, B., Hollan, J. and Cruz, I., eds. 1996. Strategic Directions in Human Computer Interaction. *ACM Computer Surveys 28* (December 1996).

Norman, D. 1986. Cognitive Engineering. In *User centered system design: New perspectives on human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum.

Perlman, R. 1976. *Using computer technology to provide a creative learning environment for preschool children*. MIT AI Lab Memo 360. Logo Memo 24 (May 1976).

Saraswat, Vijay, 1993, Concurrent Constraint Programming, MIT Press, Cambridge, MA, USA

Smith, D. C. and Cypher, A. 1999. Making Programming Easier for Children. In Druin, A., ed. *The Design of Children's Technology*. San Francisco, California, USA: Morgan Kaufmann Publishers.