

Chapter 12

Insights from Exploration of Engaging Technologies to Teach Reading and Writing: Story Baker

Michel Pahud

Abstract To engage children in learning to write, we spent several years exploring tools designed to engage children in creating and viewing stories. Our central focus was the automatic generation of animations. Tools included a digital stylus for writing and sketching, and in some cases simple robots and tangible, digitally-recognized objects. In pilot studies, children found the prototypes engaging. In 2007, a decision not to develop new hardware was made, but at today's greatly reduced tablet cost and with more capable touch and pen technology, these experiments could inspire further research and development.

12.1 Introduction

Children learn to write at different ages and with different degrees of challenge. From 2004 to 2007 our team developed and tested over 30 prototype systems designed for children 5–7 years old who have challenges in learning to write. The intent was to create highly engaging interactive experiences for children who might have limited attention spans. We made use of animation, typing, a stylus for writing and sketching, machine-recognizable tangible objects for input, and simple robot activity for viewing the outcome of actions.

Khan Academy, Sprout Online, and other sites provide learning experiences for children with real-time feedback, often focused on STEM, memory and concentration, music, and art. Writing presents unique challenges. In particular, the goal of writing is to communicate, and writing tasks that are given to children are often artificial. After our project ended, games that include writing became available. Margaret Johnson, our general manager, started a company Sabi Games through Microsoft IP Ventures to pursue further early education efforts <https://news.microsoft.com/2008/11/12/sabi-and-microsoft-launch-unique-interactive-drawing-and-education-game-for-children/>. DC Comics' "Scribblenauts" [3]

M. Pahud (✉)

Microsoft Research, 14820 NE 36th Street, Building 99, Redmond, WA, 98052, USA
e-mail: mpahud@microsoft.com

has a very large database of visual objects and actors that children can summon; only nouns and proper nouns are written.

We set out to make the computer an active partner by animating actions that a child enters as sentences, usually starting the animation when a end of sentence period is entered. We experimented with several ways for children to create simple end of sentence period. The easiest, for children who balk at a blank page, was to modify an existing story – swapping actors or adding a photo of oneself and an image of their dog to an adventure story, for example. Complete sentences result in visual animations that portray the actions described.

This requires a vocabulary comprising nouns with associated images, action verbs, and simple natural language parsing. With a limited vocabulary, we can guide children by displaying an array of available objects, with typing or writing narrowing the options. More significantly, we enabled children to add objects by inserting digital photos and sketches, making them available for use in subsequent sentences or stories. We also explored ways to enable children to add verbs. Future work could extend this, and include adjectives and adverbs.

Children's sketches are not usually polished. Some children might like to try specifying an unusual action, such as a pizza eating a dog, to see what happens. They might like to paste a head-shot photo on a story character. In consideration of such issues, we chose a non-realistic "South Park" cartoon style of animation with the concept of a mouth split for animated objects. Because highly predictable animations will not stay fresh, we developed techniques for injecting surprise and maintaining engagement. "Idle behaviors" are attached to objects – small actions through which actors appear alive and waiting to be told what to do next. Objects are given meta-data that allows a range of expression and results in less predictability; for example, a shy character flies or kicks differently than a bold one.

To encourage handwriting, one prototype rewarded small children who traced over a letter with a stylus by having images beginning with the letter appear in bubbles. Tracing a second letter popped bubbles of words that no longer fit, as the remaining images grew.

We created animated fonts through careful character-by-character analysis of how letters could stretch or distort in synchrony, allowing text beneath images to move rhythmically as a story was enacted, or emphasize words selectively. To make the fonts attractive to children, a design goal was to make them look organic and alive.

We went beyond the keyboard, stylus, and display. Cards with an image or word on one side and a scanned tag on the other could be used to construct sentences. Simple robots were used to act out story actions. We explored ways to combine visual information and robot movements to convincingly represent actions more complex than the robot could manage, such as falling down and getting up.

In addition to use in classroom contexts, we envisioned children sharing their stories online with friends and family, and a technology for remote story-telling. For example, a parent on a trip can create animated accounts of his days with photos and sketches, to be emailed to a parent at home and shared as 'bedtime stories.'

In pilot studies with children, the prototypes were well-received, but a rugged tablet was very expensive at the time. These concepts could be revisited today and extended with touch, depth cameras, and other technologies. They could be made available for younger children, and some features could find use in assistive or other settings seeking engaging captioning.

12.2 Story Baker: Will Converting Text to Animation Engage Children?

How will children react when typing ‘The dragon flew over a beach’ in a text field instantly produces a view of that activity? After a period was typed in our first prototype (Fig. 12.1), the dragon and beach appeared and the animation is launched (demo [2]). As the child types another sentence, the animation remains visible until the next period is typed (demo [2]). At that time, the animation of the new sentence replaces the previous one. To make the animations more interesting, Story Baker added background and foreground objects: For example, in Fig. 12.1b, the dragon flies behind the rock and the palm tree.

Each sentence typed creates a new animation and a new page. A series of sentences represent a story. The controls at the bottom allow navigation through the pages of the story, playing back the corresponding animations (demo [2]). At any time a child can augment the story by entering another sentence.

Our Natural Language Processing (NLP) software parses each sentence to extract the actor (noun; dragon in the example), action (verb; flew), direct object (noun; none in this example), and the location if any (noun; beach) [1]. The output is a tree containing the sentence structure. An interesting property is that different tenses, including passive tenses, have the same logical form. For example, ‘The dragon

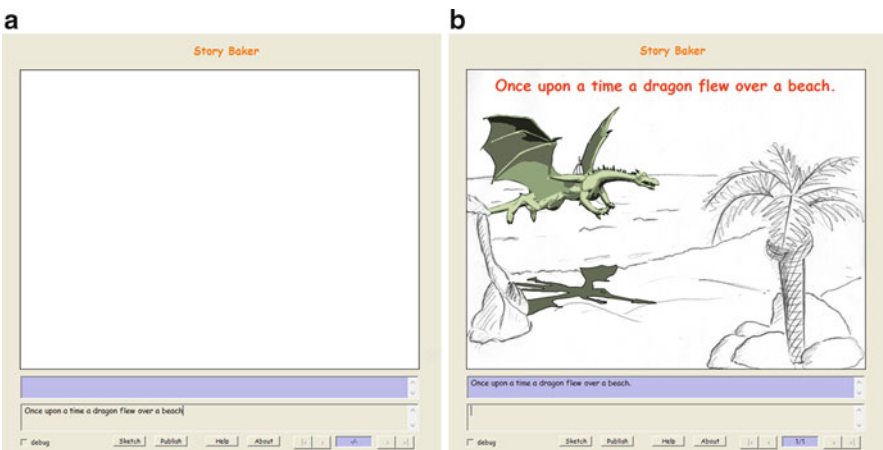


Fig. 12.1 (a) Before the child enters a sentence, the screen is blank; (b) After the period completing a sentence has been typed, the animation is launched

kicked the ball on the beach’, ‘The dragon is kicking the ball on the beach’, ‘The dragon kicks the ball on the beach’, or ‘The ball was kicked by the dragon on the beach’ will generate the same logical form and animation. This allowed the prototype to understand a wide range of sentences and also allowed us to swap languages relatively easily – we experimented with English and French versions.

We had to address word strings that did not yield a valid sentence. The intent was to display whatever we could based on the information recognized in the sentence. For example, if the program could identify only an actor, the actor was shown looking confused.

12.3 Animation Style

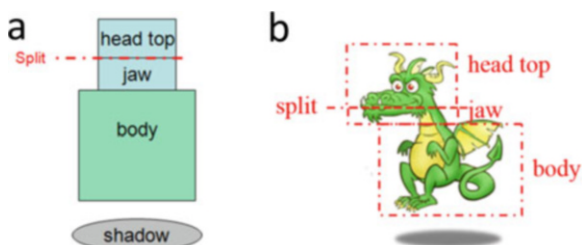
We built and explored several animation engines. One goal was to enable children to insert images or use a pen to sketch actors as they created stories, and we soon saw that a 3D engine sets expectations of perfection that would inhibit such input. Realistic 2D animations also proved limiting: ‘A man eats a ball’ could be animated, but ‘A ball eats a man’ is a valid sentence that is difficult to animate when a ball has no mouth. Our solution: simple ‘South Park Type Animation,’ in which every object is a 2D image with a mouth defined as a horizontal split line. To make it visually more interesting, an optional body with an attach point pivot on the head can rotate and scale independently. To make ‘flying’ or ‘jumping’ visually understandable, a shadow below each actor reveals whether or not it is touching the ground. Figure 12.2 illustrates this with a dragon and a generic actor with placeholders for the head and body, and a shadow.

Because all nouns are given a mouth, a ball can eat a man. After an object is eaten, it reemerges in a whimsical way, to be available for subsequent use and to avoid scaring a child if what was eaten was a person, pet, or other valued item.

12.4 Personalizing and Scaling with Sketches and Images

Children inevitably want to add objects not in our set, including photos of family, friends, pets, and toys. We enabled them to sketch objects with a stylus, insert photos, and draw over photos, with the results incorporated into the available set.

Fig. 12.2 (a) A generic actor with head, mouth, optional body, and shadow; (b) An instance of a dragon actor



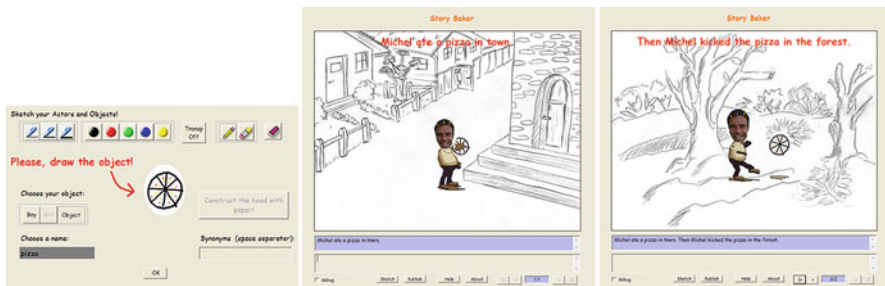


Fig. 12.3 The pizza drawing (*above*) and its use in two sentences

Fig. 12.4 Actors with photographed and sketched faces



It was engaging for children to modify the system and create unique, personalized stories this way, rather than being restricted to pre-built content. Most sketches were rough, but they were easily incorporated into the South Park style animation.

The early example above illustrates the vocabulary extension, preceding the full South Park like animation style. Encountering ‘The man ate a pizza in the town,’ with no corresponding visual for pizza, the Natural Language Processing component detects that and the child is asked to draw it (demo [2]). From then on, the system associates that drawing with the word ‘pizza’ when it appears in the text (demo [2]) (Fig. 12.3).

We created simple templates of a headless man, woman, girl and boy on which a face could be drawn or a photo placed. One appeared in Fig. 12.3; two are shown in Fig. 12.4.

A photo could be adorned with glasses, moustache, tattoos, missing teeth, and so on, if desired. Figure 12.5 shows in sequence a man’s body, after a face has been added, then when adorned with colored hair, and finally given a name (“Michel”) and used in a story (demo [2]).

We created generic activities for verbs, described in more detail in the Appendix, and considered how new verbs might be added. In Fig. 12.6, a user sketches the verb ‘cheers’ (raising both arms) on a tablet with a stylus. (This prototype also included a physical robot, described later.) To define cheer, a user stroked on the tablet to raise first the right arm, then the left.

The general issue of enabling children to create verbs is an area for future exploration. It could make use of depth cameras and other new technologies.

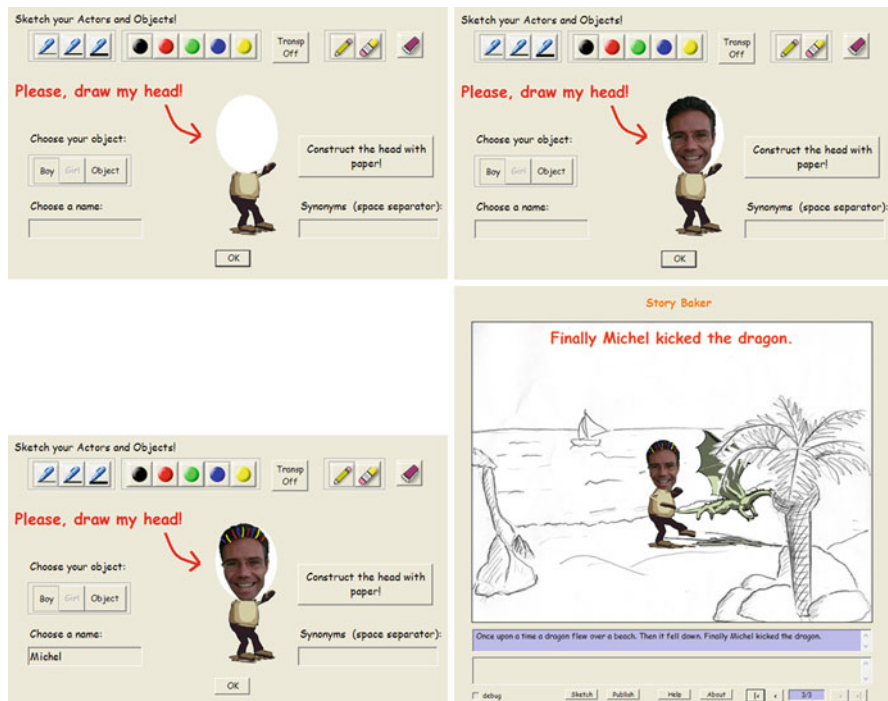


Fig. 12.5 A sketchpad showing a body as face, hair and action are added



Fig. 12.6 A user defining the verb 'cheers' by sketching

12.5 Idle Behaviors and Metadata to Add Unpredictability and Surprise

Highly predictable behaviors will cease to engage children. To counter this, we borrowed from Sims game design by assigning metadata (or 'Digital DNA') to each object to reduce predictability and inject surprise. Discovering what

different characters will do can be an incentive to keep writing stories. The metadata comprised physical characteristics (weight, strength, hardness), personality (Serious/Silly, Shy/Ongoing, Lazy/Hardworking, Grumpy/Happy, Dumb/Smart, Sleepy/Awake), and characteristics such as the horizontal split line coordinates for the mouth. They affected the visual appearance and auditory effects of actions specified by verbs. We also designed idle behaviors – small movements and sounds – that give the impression that an actor is ‘alive’ and waiting for new instructions. Idle behaviors also varied – not all actors fidgeted the same way. Manually setting up the metadata could be difficult for younger children, so the choice of character body (as seen in Fig. 12.5) was used to define the metadata. For example, when a child chooses a new character with ragged jeans or a clown suit, the character metadata is set to be more outgoing and silly than for a character given a body with a suit and tie.

The metadata constrain the resulting animation. The experience was designed to engage children even if they just riff a single sentence (in fact the first prototype to test the Southpark animation style had only one sentence). We knew that children would soon lose interest if they could predict the resulting animation upon swapping a noun. Generating a surprising animation when a word is swapped was important. For example, a child writes, “One day, a donut kicked a dragon in the backyard” and the kicked dragon follows a parabolic trajectory (Fig. 12.7). If the child modifies the sentence to be “One day, a donut kicked Michel in the backyard,” (replacing “a dragon” with “Michel” who is more “silly” than the dragon) the animation shows Michel banging against the display glass, then becoming angry and arguing with the donut (Fig. 12.8). The behavior of actors can differ, based on the metadata. A “shy” strawberry will hesitate to kick and run back and forth before deciding to kick, where an “outgoing” house will go straight at the secondary actor. Because of



Fig. 12.7 Animation of “One day, a donut kicked a dragon in the backyard.” (a) A donut approaches a dragon; (b) The dragon is kicked and follows a parabolic trajectory outside of the scene; (c) The donut celebrates



Fig. 12.8 Animation of “One day, a donut kicked Michel in the backyard.” (a) A donut approaches Michel; (b) Michel is kicked and his head whimsically hits the glass of the device; (c) Michel with a red head angrily confronts the donut

the range of combinations, even the design team had no idea what would happen when actors were swapped. In addition to audio effects, we experimented with a laugh track to accompany comical events.

For a given sentence, the animation is always the same. This enables a child to show a story to other people without being surprised, and encourages children to continue creating or modifying sentences.

12.6 Remote Storytelling

A child could create a story to share with distant grandparents. We also prototyped the opposite scenario, in which a parent on a business trip could use photos and text to easily create an animated story (basically using Story Baker as an authoring tool) to send to the family, perhaps as a bedtime story. In the prototype, on the receiving side, a series of sentences would be projected on a wall to be read aloud by a parent or the child. When the speech recognition technology detected sentence completion, the corresponding animation was triggered. The animation could quickly be constructed by the traveler using selfies, pictures taken during the day, and other materials at hand. For example, a parent travelling to Paris for business could take a selfie with the Eiffel tower and make a quick cartoonish story to share with family at home.

12.7 Teaching Handwriting Through a Delightful Stylus Experience

Handwriting has significant educational advantages over typing [8, 9]. Some children take to it easily at a young age, others can use an incentive to work on it. We looked into recognition of free-form handwriting of kids learning to write to see if it could be used as input in story creation, but it is too inaccurate. For the youngest children, starting with a blank page is intimidating and yields many invalid sentences. Instead, we provided them with story templates that they could modify by swapping words, then seeing the new sentence that they had created acted out. To specify a word, they used a traditional letter-tracing task made more engaging through animation.

As the child traces, small animations (called ‘particles system’) appear to emerge from the stylus. In Fig. 12.9, where letters are grouped as in the familiar “Now I Know my A-B-C’s” song, as the child traces the letter ‘b’, images of ‘banana’, ‘belt’, ‘bee’, ‘book’, and ‘broccoli’ appear in bubbles (demo [2]). If the child next traces the letter ‘a’, the bubbles for ‘belt’, ‘bee’, ‘book’, and ‘broccoli’ will pop, leaving the visual ‘banana,’ which grows, its size inversely proportional to the number of letters remaining. The child can continue, or can pop the bubble with a quick stylus touch

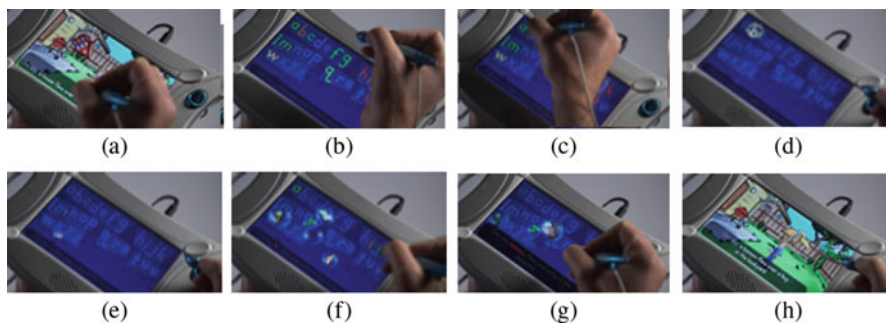


Fig. 12.9 (a) The animation plays before the child selects a word to replace; (b) The alphabet for tracing; (c) ‘b’ is traced, animates (d) and falls into place (e); (f) the visuals for ‘banana’, ‘belt’, ‘bee’, ‘book’, and ‘broccoli’ are visible in bubbles, with continuation letters highlighted; (g) a child pops the ‘banana’ bubble, the word is inserted and a new animation plays (h)

at any time to insert the word; if an entire word is traced, the bubble automatically pops and the word is inserted.

If a great many letter-matching choices are available for a given combination of letters, the bubbles could rotate over time to keep a manageable number visible at one time. Having to trace additional letters to constrain the set could motivate additional writing. When a hovering stylus can be sensed, bubbles could gently move aside to leave space for the child to trace the letter underneath.

12.8 Classroom Presenter

Because children having trouble learning to read and write may not be motivated by text in general, we considered how to make fonts themselves lively and engaging. Our goal was a “font with feelings,” which we termed “Fontlings.” Canned animations require extensive artistic work and do not scale to other font types. Previous work on parametric font modification [6] and kinetic typography [4, 5, 11] did not meet our needs. The former was designed for a different purpose; it was too CPU intensive for our prototype hardware and would be difficult for children to work with. The latter didn’t convey organic, living sense that we sought for this experience. To achieve a simple, procedurally animated, lively font, we decided to adapt our South Park-like animation. The letter ‘h’ split above or below the horizontal line as with objects could look strange and perhaps not be recognizable. Instead, we created a join that connects the top and bottom part of each character, identifying a rotation point, as shown in Fig. 12.10 for the letter “f”. Each part can move, rotate and/or scale independently while connected, yielding very interesting animations. Letters might have multiple joins; for example, for the letter “h,” the split was set close to the base of the letter, so two joins connect the parts.

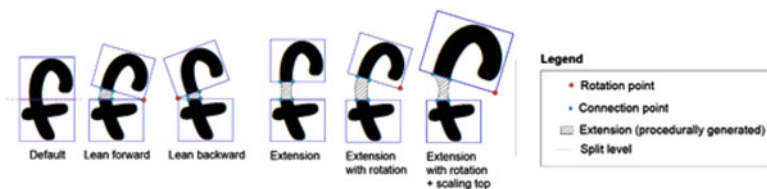


Fig. 12.10 A dynamic join connects parts of a character when animated

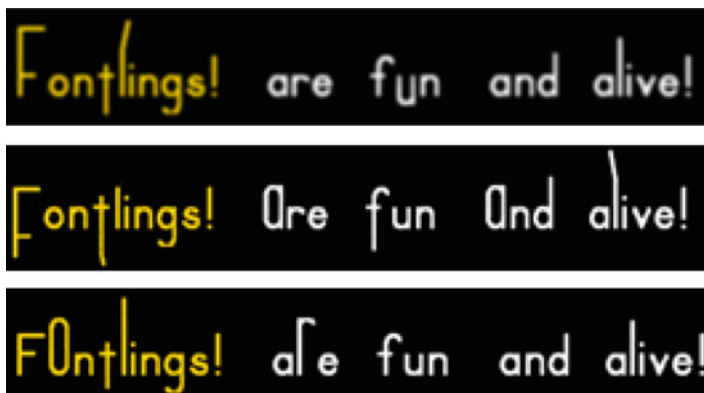


Fig. 12.11 Snapshots during idle behavior animation, illustrating the organic and lively impression

One interesting point to note with this approach is that a font type only need the split horizontal line height information for each character once for all. If someone wants to support other font types in the future, they just have to define a split line for each character for this new font type.

We implemented a fully working prototype, defined a scripting language, and wrote scripts to animate idle, happy, and misunderstood behaviors. The results impressed viewers (demo [2]). Especially appealing were a worm-like organic feel of idle behaviors and a rubber-band effect for misunderstood behavior. Figure 12.11 shows snapshots of idle behavior, although imagining the full effect is not easy. For more technical details, the data structure for Fontlings appears in the Appendix.

Within the context of Story Baker, Fontlings were used to connect images and text during interactions (Fig. 12.12). Touching the image of the dragon produces a dynamic link from the image to the word “dragon,” and the font on the word becomes animated (Fig. 12.12a). Touching the word dragon animates it and produces a visual effect on the dragon image (Fig. 12.12b). The sentence exhibits idle behavior most of the time, but becomes fully playful and excited from time to time (Fig. 12.12c). This experience is enhanced with audio accompaniment. (Demo [2])

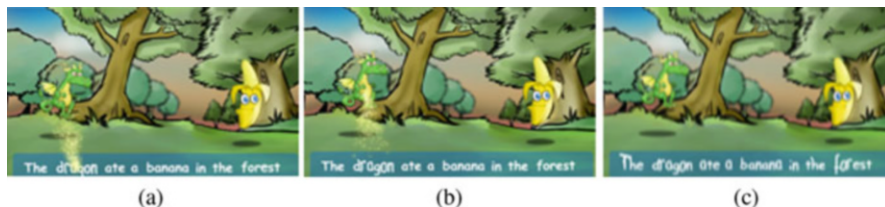


Fig. 12.12 Within the context of Story Baker, Fontlings were used to connect images and text during interactions. (a) Touching the dragon image produces a visual link to the word. (b) Touching the dragon word produces a visual link to the image. (c) The sentence occasionally animates spontaneously



Fig. 12.13 The “dragon ate Michel in the town”. (a) A card with a tag (*circled*) that is recognized by the table. (b) Animation based on a sentence; cards are enhanced to be more visible

12.9 Incorporating Real-World Objects

We prototyped a simpler story construction interface that involved physical cards, each with a word or set of words, with a code recognizable by a Microsoft PixelSense [10] table on the back (demo [2]). The cards represented nouns, verbs, and locations. A child who could read the words could use them to construct a sentence to be animated, such as “the dragon ate Michel in the town.” (Fig. 12.13.)

We experimented with the use of robot actors. With a robot positioned next to the tablet, when a story reached the word “robot,” a virtual robot appeared on the screen and the specified action affected both the virtual and real robots. We also created an verb authoring with experience on a tablet with a stylus described earlier that optionally animated a robot when a new verb was authored (Fig. 12.6). We used a ROBOSAPIEN from WowWee [13] and built a robotic API to send infrared commands to the actual robot. Figure 12.14a shows a robot that was kicked by the virtual actor in response to the sentence, “Then Michel kicked the robot on the beach.” We explored the use of text to instruct the robot to act on physical objects, within the limitations of what the robot can do. In Fig. 12.14b, the robot has responded to the sentence “Once upon a time, a robot grabbed a cup.”



Fig. 12.14 (a) A robot that was just kicked by a virtual actor. (b) A robot that just picked up a cup

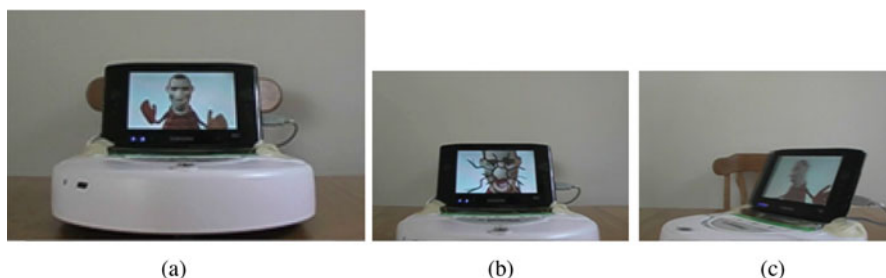


Fig. 12.15 “One day CoolBot was at home. CoolBot was very happy and started dancing. Suddenly he tripped and fell down. Finally, he stood up and started spinning”. (a) CoolBot dances: the physical robot moves back and forth while the virtual character waves its arms;. (b) CoolBot falls. (c) CoolBot spins

We also positioned a display on an armless Roomba from iRobot to ‘symbiotically’ act out verbs, working to overcome some limitations (demo [2]). The Roomba had a UMPC (Ultra-Mobile PC) with a touch screen controller. To ‘dance’ the robot moved back and forth while the virtual character waved its arms (Fig. 12.15a). ‘Fall down,’ ‘fly,’ and ‘eat’ were more challenging. For ‘fall down,’ the robot moved forward very fast while the virtual actor screamed, then stopped with an image appearing that seemed to be the screen cracked by the character’s head (Fig. 12.15b). For ‘fly,’ the virtual character could extend its arms like wings as the robot moved. To ‘eat,’ the robot could suddenly move forward as the display showed an object moving into the virtual character’s mouth.

One story was “One day CoolBot was at home. CoolBot was very happy and started dancing. Suddenly he tripped and fell down. Finally, he stood up and started spinning.” Figure 12.15 shows the robot ‘dancing,’ ‘falling down,’ and ‘spinning.’ In our simplified prototype, robot stories could not be authored, but the UMPC touch display would support it, perhaps by stylus tracing as described above.



Fig. 12.16 (a) Two robots, physical props (*trees*), a virtual ball, and a virtual shark hidden in the lake; (b) Acting out “Roby kicked a ball.” (c) After acting out “A big shark upset kicked the ball very strongly toward Robo and made him fall down”

12.9.1 Robotic Playset

We also explored a robotics playset, which provides children with a starting point that constrains what they say while retaining a sense of freedom (demo [2]). The robots were programmatically controlled with infrared signals dynamically sent from the PC controlling the PixelSense table, synchronized with the animation displayed on the table (Fig. 12.16). For this prototype we provided the cards with words and tags to use in authoring stories.

12.10 Pilot Studies

12.10.1 Free-Form Study

In pilot studies of early versions of Story Baker, children typed free-form text using the prototype shown in Figs. 12.1, 12.3, and 12.5. These were undertaken to get feedback and guide or inspire design changes, and are described anecdotally based on review of videotapes. In general, children were fully engaged and did not seek a perfect experience. Rather than a defect, having to draw missing objects was seen as “teaching the computer”; personalizing the content was a positive experience.

12.10.2 Surprising Animation Study: Single Sentence

A subsequent study explored reactions to the South Park style of animation in which tracing was used to enter text, an early version of the prototype shown in Fig. 12.9 with the kind of surprising animations such as shown in Fig. 12.7 and 12.8. We had children adapt a sentence that they were given to see the effect of animation that varied based on digital DNA. As a child modified the text in a sentence with a simple word selection interface, others watched and made suggestions (“Do, ‘the microscope kicks the house’”), and they laughed at unexpected animations. A

parent reported that the children talked about it on their way home. With variation, even a single simple sentence could be highly engaging.

12.10.3 Extended Study Session 1: Modifying South Park Style Stories

In a more extended study, six children over two sessions with a mature version of Story Baker examined word entry by tracing with six stories (median length 9 sentences). The primary actor, secondary actor, action and location could be modified. Children could choose among realistic, space travel, and fairy tale story themes. The prototype had 13 actions,¹ 76 actors,² and 16 locations.³ Each actor had a metadata file that defined physical and personality characteristics inspired by Snow White's dwarves.⁴ The actors, actions and locations were inspired by a list of words familiar to children in the target age range.

12.10.4 Extended Study Session 2: Personalization by Inserting Pictures of Children, Family, Pets, and Toys

Personalization was introduced in the second session, a week after the first: parents were asked to provide pictures of their child, one or more parents or guardians, any siblings or pets, and preferred toys. We placed them in the system, using 'dad' and 'mom' for parents, with faces on a cartoon body. For children we used their names and put their faces on boy or girl cartoon bodies. We defined a split line for a mouth

¹List of actions (verbs) initially on Story Baker prototype: chase, create, dance, eat, fall, fly, jump, kick, love, marry, meet, play, and see.

²List of actors (nouns) initially on Story Baker prototype: Alligator (an), apple (an), astronaut (an), bag, ball, banana, bee, beehive, belt, bicycle, book, boy, broccoli (some), cake, car, cat, chair, cheese (some), clock, clown, cookie, couch, cow, dinosaur, dog, donut, dragon, duck, fish, girl, Gus (actor name), hand, hippo, horse, house, invention (an), Jack (actor name), Jish (actor name), kangaroo, King, lasergun, lemon, letter, lion, mailbox, man, Michel (actor name), microscope, monkey, mouse, pancake, pants (a pair of), penguin, phone, pig, pizza, plane, prince, princess, queen, racecar, Red (actor name), robot, rock, rocket, shark, shoe, snake, strawberry, table, toy, tree, truck, turtle, underpants (a pair of), and woman.

³List of locations initially on Story Baker prototype: backyard, beach, bedroom, castle, classroom, den, factory, farm, forest, kitchen, moon, park, party, store, town, and zoo.

⁴Physical: Weight [0..9] (0 – Light, 9 – Heavy), Strength [0..9] (0 – Weak, 9 – Strong), Soft/Hard [0..9] (0 – Soft, 9 – Hard). Personality: Type Human, Animal, Vegetable, Mineral, Serious/Silly [0..9] (0 – Serious, 9 – Silly), Shy/Ongoing [0..9] (0 – Shy, 9 – Outgoing), Lazy/Hard worker [0..9] (0 – Lazy, 9 – Hard worker), Grumpy/Happy [0..9] (0 – Grumpy, 9 – Happy), Dumb/Smart [0..9] (0 – Dumb, 9 – Smart), Sleepy/Awake [0..9] (0 – Sleepy, 9 – Awake).

that was about the height of the actual mouth. For the pets and toys, we didn't attach the picture to a body, but defined a split line for the mouth.

Animating sentences in the virtual world engaged children, even or especially those making no sense in the real world. A child excitedly repeated "The pizza ate the fish," turning the device to show us his animation. Personalization was popular. An excited child said, "I got daddy", "I got mommy and daddy" and later "it was super fun" and "yeah!" when asked if she would do it again. One child laughed at length after making a rocket fly over his dad in the backyard. Surprised to see her pet in a bubble, one said "Oh <pet name>, cutie!" before creating a story in which her pet ate a fish in the zoo. Another said "Where is dad?" after tracing the letter "d," then "Here is daddy!" and giggling when seeing his face in a bubble. Other examples were "<child name> flew over dad in the castle." Asked about the experience, one said, "I like it very very very, I love it."

The decision not to continue was based on issues around producing a successful device given the hardware available in 2007. (Specifications are in the appendix.) The reactions to the prototypes was very positive, and a decade later, with costs down and capabilities up, these ideas merit reconsideration.

12.11 Conclusion

Over four years, we explored how to engage children in writing stories sentence by sentence, with each completed sentence instantly converted to an animation. We found that 'realistic' animations set high expectations and made many sentences difficult to animate. Our South Park type of animation yielded more whimsical and versatile animations and invited children to extend the vocabulary by sketching and annotating. We explored animating robot actors alongside the virtual world, and enlivening stylus tracing of letters with relevant animations. Our work is a step toward teaching literacy. Opportunities that seem worth exploring follow.

- Our prototypes limited sentences to two actors, one verb, and one location. This should be extended to more complex sentences, with more actors, adverbs and adjectives. For example, "The big dragon and small bird quickly ate the blue banana." For more advanced children, we could animate more complex sentences.
- We enabled new verb creation through the use of stylus strokes to 'push' limbs of a virtual robot (which also animated a real robot). Another approach might be acting in front of a depth camera. Variants of a verb for different digital DNA could be authored the system could prompt: "Please act out someone cheering shyly; now, act out someone cheering in a goofy way." The system could create variants for different levels of shyness, goofiness, and so on. Verb authoring with multi-touch and tracing trajectories could be explored.
- With the trend toward large pen-sensitive and multi-touch displays, with different devices working together, one could explore collaborative story creation. Multi-

ple children might co-author on a large display, or remotely via smartphones or tablets. Every child might contribute to a portion of the story, or small groups could make and share their stories. Children could draw sets of missing nouns on a large display.

Appendix

A.1 *Prototype Tablet*

In 2007, we made a prototype device with a 7" 800×480 pixel display (Fig. 12.17a) and 4 GB solid state memory, running Windows CE on an AMD Geode 500 MHz processor. It had a FinePoint digitizer with an active stylus tethered by a retractable power cord. The device was designed to survive 3 foot drops. Its overall size was $11 \times 6.25 \times 1.4$ inches, weighing 0.8 kg. By design the device looked like a painter's palette to emphasize the stylus and creativity experience. The device also had a handle to make it easy to carry. The stylus was tethered to avoid losing it and to provide power without requiring a battery (Fig. 12.17a). A polished version of Story Baker was implemented on the device (Fig. 12.17b).

A.2 *Sentence Structure*

Figure 12.18 shows the sentence structure with some of the objects and actions. Sentences have a primary actor, action, a secondary actor, and optional background. We created an API to convert and animate a sentence (see Visual Sentence API documentation [12]).

The top row of Fig. 12.19 illustrates a generic eating animation. Beneath it is an instance of a dragon eating a banana: From left to right, the dragon faces the banana, its mouth opens, the banana enters its mouth, and the dragon chews. Chewing uses rotation and scaling, and can dynamically recolor the character; for example, the



Fig. 12.17 The prototype hardware (2007). (a) The prototype device. (b) Story Baker running on the device

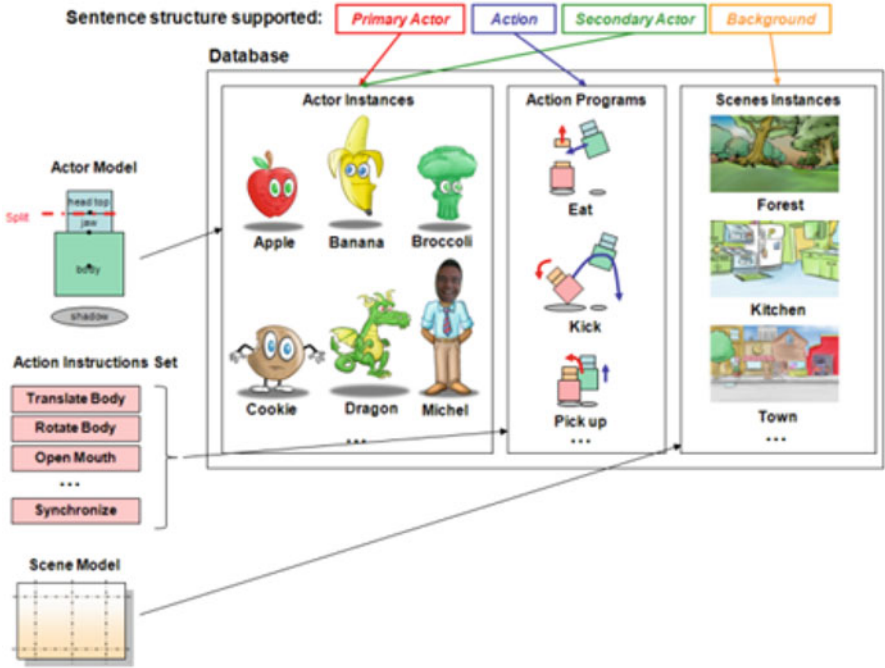


Fig. 12.18 Animation system overview

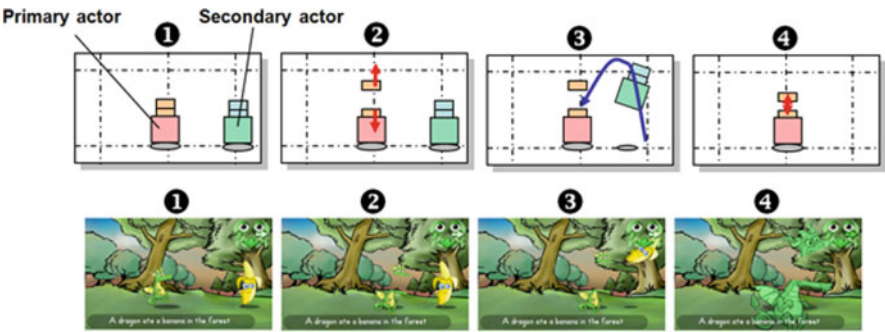


Fig. 12.19 Animating a dragon eating a banana in the forest. Top: Generic eating animation, Bottom: Animating a dragon eating a banana in the forest

head and/or body could turn green to depict disgust or overeating. After chewing, the primary actor ejects the food to avoid trauma if the secondary actor is a favorite character. The animation is accompanied with audio to make it convincing.

Figure 12.20 illustrates how our system transforms a sentence into an animation. The digital DNA, image, and audio of the primary and secondary actors are drawn

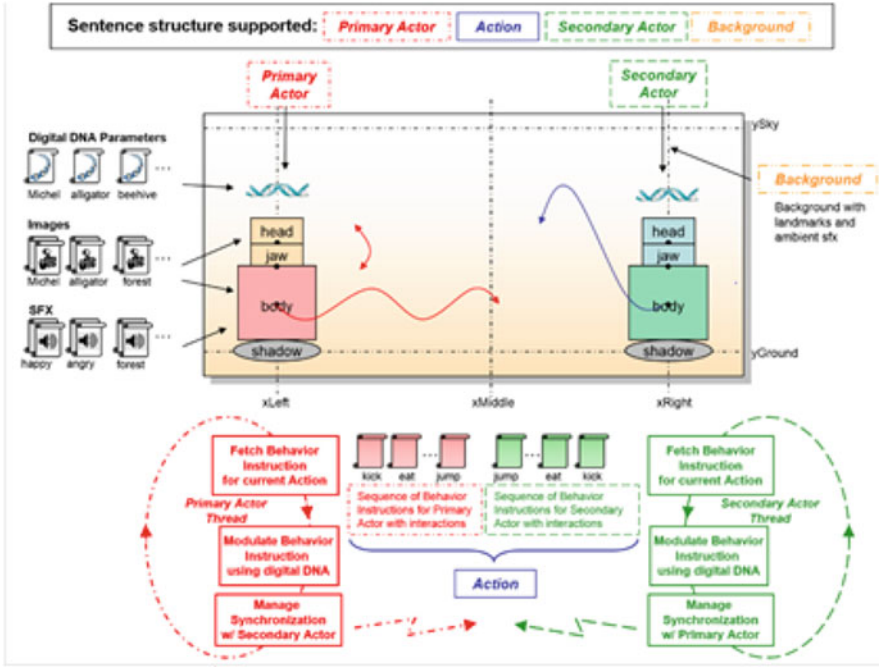


Fig. 12.20 Model for animating with digital DNA variations

from the KidWords dictionary. Behavior instructions are fetched for each actor and the resulting trajectories and sounds are modified to accord with their meta-data. A synchronization barrier ensures that the actors synchronize. For example, an actor that is kicked should not begin flying off before the kick is delivered. Since the primary actor will likely have different digital DNA than the secondary actor, they could behave quite differently (e.g., a “silly” primary actor might stumble around before kicking). Actions are defined as a sequence of behavior instructions, each of which indicates what the actor is to do, but not how: That is defined by the digital DNA. For instance, a digital instruction to move from point A to point B could be executed by striding, stumbling, or hopping. For more details, see this list of macro actions [7].

Figures 12.21 and 12.22 show generic animations for “kick” and “pick up” actions, respectively, without variants due to digital DNA.

A.3 Fontlings Data Structure

In the data structure for Fontlings (Fig. 12.23), each letter (or number/special character) has a split line and a rotation point for each part (top and bottom part). Each letter has position, angle (rotation), and scaling parameter (not shown) for

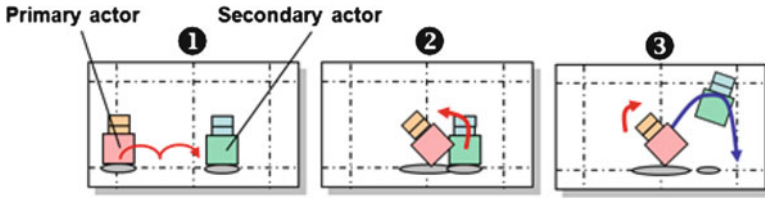


Fig. 12.21 Example of kicking

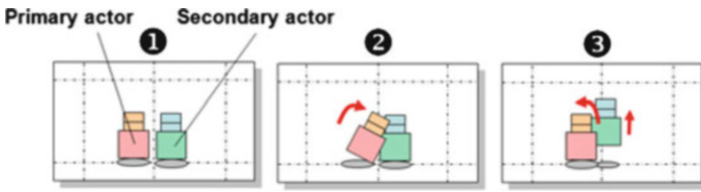


Fig. 12.22 Example of picking up

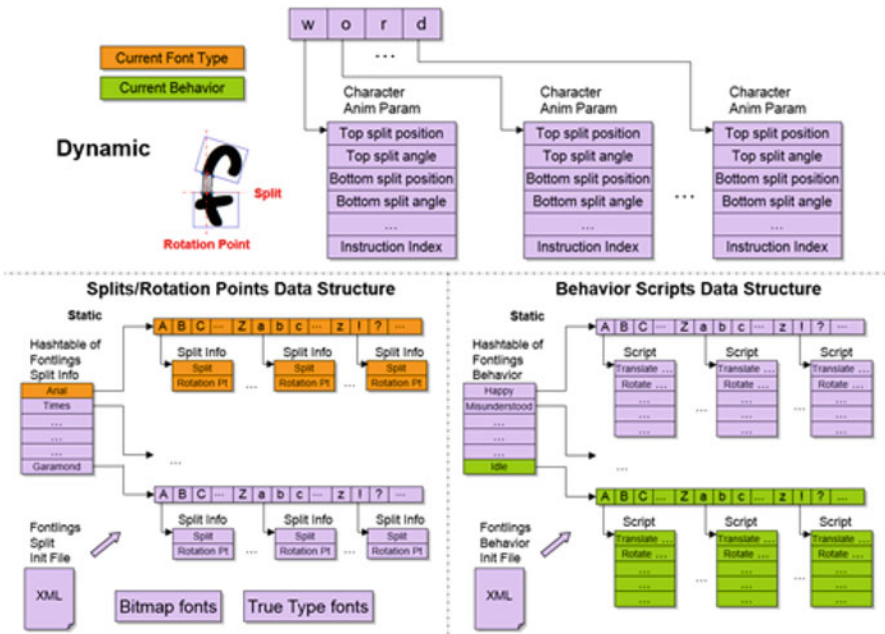


Fig. 12.23 Fontlings data structure

each part, and each letter animates independently (Top split position, Top split angle, Bottom split position, Bottom split angle, etc. in Fig. 12.23). An instruction index (Instruction Index in Fig. 12.23) tracks its position in the execution of the script. One letter or word could be in idle mode while another is happy. In this way, a specific word can highlight something while other words are alive but less dramatic. For each letter, scripts define behaviors (idle, happy, misunderstood, etc.). The Splits/Rotation Points Data Structure refers to a hash table with an entry key for each font type (Arial, Times, etc. in Fig. 12.23). It specifies rotation points and the height of the split line as a percent of the character's height. The Behavior Scripts Data Structure is a hash table with the behavior name as entry key (Happy, Misunderstood, Idle, etc. in Fig. 12.23). It contains a script for that behavior for each character. Figure 12.23 also highlights a specific example for the font type Arial (in orange color) and the behavior Idle (in green color).

The data structures are initialized by the xml file Fontlings Split Init File, which contains the height of the split line for each character of a given font, and Fontlings Behavior Init File, which contains the scripts to animate each character for each behavior (see bottom of Fig. 12.23).

Acknowledgements This work would not have been possible without Chuck Thacker, Craig Mundie, Margaret Johnson, Takako Aikawa, Lee Schwartz, Bill Buxton, Jonathan Grudin, Howard Phillips, Tom Steinke, Duncan, John D. Bransford, Diana Sharp, Heinz Schuller, Margaret Winsor, Allison Druin, Tim Kannapel, Arin Goldberg, Keith Daniels, Xiang Cao, Anoop Gupta, Chris Quirk, Bill Dolan, John Paquin, Neal Noble, Chad Essley, Lindsey Sparks, John Manferdelly, Xiaolong Li, the children participating in studies, and perhaps others I forgot to list.

References

1. Aikawa, T., Schwartz, L., Pahud, M.: Nlp Story Maker (2005)
2. Cptte demo page. <https://aka.ms/mpahud-CPTTE2016-demos/>
3. Dc Comics' "scribblenauts". www.scribblenauts.com
4. Lee, J.C., Forlizzi, J., Hudson, S.E.: The kinetic typography engine: an extensible system for animating expressive text. In: Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology, pp. 81–90. ACM (2002)
5. Lee, J., Kim, D., Wee, J., Jang, S., Ha, S., Jun, S.: Evaluating pre-defined kinetic typography effects to convey emotions. *J. Korea Multimed. Soc.* **17**(1), 77–93 (2014)
6. Loop, C., Blinn, J.: Resolution independent curve rendering using programmable graphics hardware. In: *ACM Transactions on Graphics (TOG)*, vol. 24, pp. 1000–1009. ACM (2005)
7. Macro actions. <https://aka.ms/mpahud-CPTTE2016-MacroActions/>
8. Mueller, P.A., Oppenheimer, D.M.: The pen is mightier than the keyboard advantages of longhand over laptop note taking. *Psychol. Sci.* **25**(6), 1159–1168 (2014)
9. Oviatt, S.: The design of future educational interfaces. Routledge, New York (2013)
10. Pixelsense table. <https://www.windowcentral.com/microsoft-surface-pixelsense-table>
11. Strapparava, C., Valitutti, A., Stock, O.: Dances with words. In: *IJCAI*, pp. 1719–1724 (2007)
12. Visual sentence API documentation. <https://aka.ms/mpahud-CPTTE2016-VisualSentenceAPI/>
13. Wowwee. <http://wowwee.com/products/robots>