
This item was submitted to [Loughborough's Research Repository](#) by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

Storyboarding: a method for bootstrapping the design of computer-based educational tasks

PLEASE CITE THE PUBLISHED VERSION

<http://dx.doi.org/10.1016/j.compedu.2008.01.002>

PUBLISHER

© Elsevier

VERSION

AM (Accepted Manuscript)

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Jones, Ian. 2019. "Storyboarding: A Method for Bootstrapping the Design of Computer-based Educational Tasks". figshare. <https://hdl.handle.net/2134/8610>.

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Storyboarding: A method for bootstrapping the design of computer-based educational tasks

Ian Jones

Abstract

There has been a recent call for the use of more systematic thought experiments when investigating learning. This paper presents a storyboarding method for capturing and sharing initial ideas and their evolution in the design of a mathematics learning task. The storyboards produced can be considered as “virtual data” created by thought experiments that provided an audit trail of evolving design ideas and enabled peer feedback to be sought. The outcomes were a diagnostic computer-based task for investigating mathematical thinking and the development and analysis of the storyboarding method itself. It is argued that storyboarding offers an opportunity for systematising thought experiments and peer feedback towards more robust learning task designs and theoretical conjectures.

Keywords: Authoring tools and methods; Elementary education; Human-computer interface; Interactive learning environments

1. Introduction

This paper describes a method developed and used to bootstrap design ideas for a computer-based mathematical learning task. Storyboarding offered an alternative approach to sketching interface schematics and enabled design ideas to be fleshed out for prototyping and pilot trialling. The method allowed a variety of factors to be captured, documented and shared including the design of software resources, learners' interaction and collaboration, and the role of participant researchers during trials.

The term "bootstrapping" as used here refers to the development of early, rough ideas for learning tasks towards prototyping (Pratt, 1998). It is a creative process that involves thought experiments about imagined learning tasks. Thought experiments draw on the existing educational literature, the curriculum domain of instruction and the designer's professional intuitions in order to develop resources and tasks to support learning. This process typically begins by considering a real-world mathematical context that is culturally familiar to children, such as dice games or calculators. Pratt (1998) described the process of bootstrapping an intervention into children's mathematical thinking as follows:

"I played mind games in which I imagined stereotypical children, who were presumably amalgams of all the children I had worked with in the past, using hypothesised tools. The outcomes of such virtual experiments were roughly hewn sketches of what sorts of tools might address the strands of knowledge identified from the epistemological analysis" (p. 107).

Thought experiments have proved invaluable for bridging the gap between mathematics curricula, educational theory and the localised implementation of educational interventions. However, they have been acknowledged to lack the rigour and systematicity of actual experiments. Papert (2006) recently called for "a more developed culture of properly disciplined thought experiment [in order to] enrich thinking about learning" (p. 582) but offered no explicit suggestion of how this might be achieved.

In this paper, storyboarding is offered as a method for disciplining, capturing and making accountable the bootstrapping phase of task design. The next section outlines the background to the study, early failed attempts to sketch interface prototypes and the subsequent need for storyboarding. Following this, the tool (MS *PowerPoint*) and technical procedures for creating storyboards are set out. The storyboards produced are then described and analysed in order to trace how both the task and the storyboarding method itself evolved and solidified in response to peer feedback. The central themes are a shift of focus away from interface functionality and on to collaborative learning and the role of participants. Finally, the method and outcomes are reflected on to highlight the scope, strengths and pitfalls of using storyboards to bootstrap task designs.

2 Background

This section presents the background and rationale for the development of the computer-based arithmetic task and storyboarding method used to implement it.

2.1 *The role of the equals sign in learning arithmetic*

Most pupils struggle with symbolic arithmetic at school. Even those who do well in assessments of procedural skills often lack a conceptual understanding of the

mathematics involved (DfES, 2002). The equals sign is an often cited case. Most primary and early secondary pupils are familiar with reading and using the equals sign, but few show awareness of its equivalence meaning and instead interpret it as a place-indicator for the result of a calculation (Behr, Erlwanger & Nichols, 1976). A relational understanding of the equals sign in terms of equivalence has been shown to correlate with pupils' flexible use of efficient arithmetical strategies (Molina, 2006) and algebraic equation solving skills (Knuth, Stephens, McNeil, & Alibali, 2006).

2.2 A rough design idea for an arithmetical learning task

The method reported here comes from a study into the design principles for engaging learners with a relational understanding of arithmetical equivalence. The starting point was a previous small study in which a variant on the simple school calculator (the *Equivalence Calculator*, Figure 1) was designed and trialed to investigate conceptions of the equals sign. The findings had confirmed those reported in the literature: the pupils worked readily with the equals sign but appeared not to grasp its relational meaning (Jones, 2006).

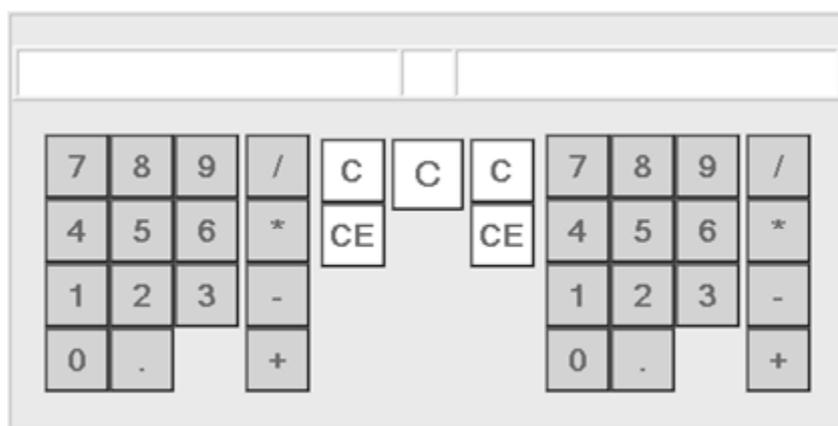


Figure 1: The Equivalence Calculator

A way forward was offered by a didactical technique (Hewitt, 2003) for relating pupils' natural language arithmetic, as in "add 3 to 6 and then times by 2", to formal notation, as in $2 \times (6 + 3)$. The initial rough idea was that pupils might verbalise natural language calculation strategies (as in "83 plus 45 is 80 plus 40, which equals 120, and 3 add 5, and that's 8, which makes 128") and then input them at a keyboard as formal notation (as in, $80 + 40 = 120$; $3 + 5 = 8$; $120 + 8 = 128$) in order to program their strategies into a computer. These strategies could then be played out as animations and shared with classmates. Collaborative games arising from this were imagined, such as "guess my strategy".

2.3 Initial software schematics

The initial rough design idea needed fleshing out and testing for feasibility. This was attempted by sketching interfaces for inputting and transforming arithmetical notation. Vaguely, notation would be inputted at the keypad and appear on screen and some buttons for processing (partitioning, commuting, composing and so on) would be provided. Every time a line of notation was processed it would scroll up for the next line. However it was difficult to imagine a coherent design that could conceivably be implemented as a working prototype. It seemed the quirks and assumptions of pupils' typical natural language explanations would be too removed from the precision and explicitness required by the computer to process them. Attempts to bridge this gulf by

elaborating the software's functionality led to interface sketches that were evermore unwieldy and restrictive (Figure 2).

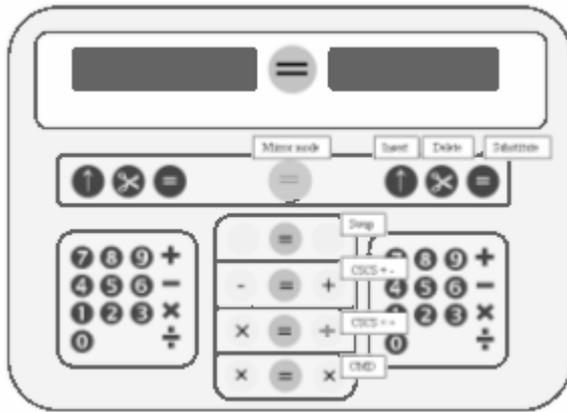


Figure 2: Early interface schematic

2.5 Origins of the method reported in this paper

Much of the content of the initial thought experiments involved imagining the arithmetical notation learners might construct and transform, without consideration of interface features and functionalities, and this was not being captured directly. Instead these emerging ideas were being embodied in the buttons and display areas of increasingly complicated interface schematics. In light of this an alternative approach was experimented with: making slides in which an epistemic child, C, inputted arithmetical notation, and representing C's stepwise transformations of notation from slide to slide. This developed into a trial-and-error storyboarding method that itself evolved. The method involved iterative refinement in which each storyboard was reviewed and modified when producing the next.

2.6 Storyboarding in HCI and education literature

Software storyboarding has its origins within the realm of human-computer interaction (HCI). In a manner analogous to film production, "the use of storyboards in the development of computer systems is a way to 'sketch out' the future system early in the development process" (Madsen & Aiken, 1993, p. 57). There are two main attractions to the storyboarding method for interface developers: (i) the involvement of users as co-developers, rather than in their traditional role of reviewers; and (ii) accountability in the form of an "audit trail of the requirements analysis process" (Madsen & Aiken, 1991, p. 39).

Within education, storyboarding has been used to design for online learning. However, it tends simply to replace technical documentation for software rather than specify the instructional design process (Jantke & Knauf, 2005). Storyboarding methods that focus more explicitly on learning processes have been developed for the design and implementation of undergraduate courses (Knauf, Sakurai, Dohi, Tsuruta & Gonzalez, 2007), and have been explored as a learning tool for pupils (e.g. Barron, 2000). However, there has been very little reporting of storyboarding methods being systematically applied to the design of domain-specific learning tasks.

3. Method

The purpose of this section is to provide an overview of the tool (MS *PowerPoint*) and technical procedures used to create the storyboards. In later sections the method's potential and limitations are reflected on.

3.1 Tool and procedures

The tool used was Microsoft *PowerPoint*. *PowerPoint* offered several benefits, not least of which is the practical fact that it was already installed and is easy to use. Simplified graphical images of resources could be created and then, using the copy and paste tools, new slides cloned and small stepwise adjustments made to the graphics. When design issues and problems were encountered an earlier slide was returned to, copy-pasted anew, and an alternative possible trajectory of events considered. Indeed the copy and paste tools, and the speed by which a new slide could be made by a minor adjustment to a previous slide, was a key benefit of using the computer rather than sketching by hand. The “click to add notes” facility allowed slides to be annotated as appropriate. These annotations were date-stamped to build up a time-line of evolving ideas, theories and conjectures.

Another key benefit of the computer is that electronic media are more portable and reproducible than paper. The commonality of *PowerPoint* ensured portability and .ppt files were shared with a colleague who added his own date-stamped comments to relevant slides to provide feedback and offer suggestions. Indeed familiarity, ubiquity and file portability are the key features that made *PowerPoint* preferable to specialised storyboarding software despite many such packages being freeware.

Nonetheless, there is a major drawback to using *PowerPoint*. Unlike Microsoft *Word*, *PowerPoint* does not offer a “Reviewing” mode and comments had to be manually dated and formatted in order to be findable within storyboards that quickly grew to dozens of slides. Also, the drawing tools are constrained to basic shapes and text. It would not do to import more sophisticated images created elsewhere because this would lead to a loss of control when making slide-by-slide adjustments. On the plus side, however, the constraints of the drawing tools do force simplicity and this in turn speeds the replication and minor adjustments needed to create a new slide.

3. Virtual data

The storyboards produced by the above method can be considered as “virtual data” captured from thought experiments. This section reports on these virtual data and describes the design issues that arose as a consequence of producing them.

3.1 Quantitative overview.

The bootstrapping of the computer-based task occurred over a period of six months during which time a total of ten storyboards were produced. A storyboard was abandoned and a new one started every time the design appeared to run into problems. The ten storyboards, then, can be considered to be ten iterations of thought experiment over the six month period. In early storyboards, design ideas changed substantially over each iteration but converged over the final few iterations. This is reflected in the discontinuous contrasts across the early storyboards and the stabilised continuity of the final three storyboards. Nonetheless, certain themes play out repeatedly across all ten storyboards, and these are teased out in following sections.

The ten storyboards varied in length from 15 slides to 170 slides each and comprised just over 600 slides in total. The earlier storyboards tended to be longer and the later storyboards tended to be shorter, again reflecting iterative convergence upon a final design.

Each storyboard comprised one or more scenes, separated by distinctive clapper-board slides to indicate a “cut”. Each scene correlated, more or less, with C working

on a particular problem such as programming a calculation strategy for 18×8 . Successive scenes in a given storyboard would typically follow alternate hypothesised strategies for the same problem, or a similar strategy applied to different problems.

A quantitative summary of the storyboards produced by the method is shown in Table 1. The filename prefix “VT” stands for “virtual trial”.

Filename	Number of slides	Number of scenes
VT1.ppt	85	6
VT2.ppt	170	9
VT3.ppt	73	5
VT4.ppt	42	2
VT5.ppt	59	2
VT6.ppt	52	3
VT7.ppt	15	1
VT8.ppt	42	4
VT9.ppt	50	4
VT10.ppt	23	3
TOTAL	611	39

Table 1: Quantitative summary of the storyboards

3.2 Qualitative overview.

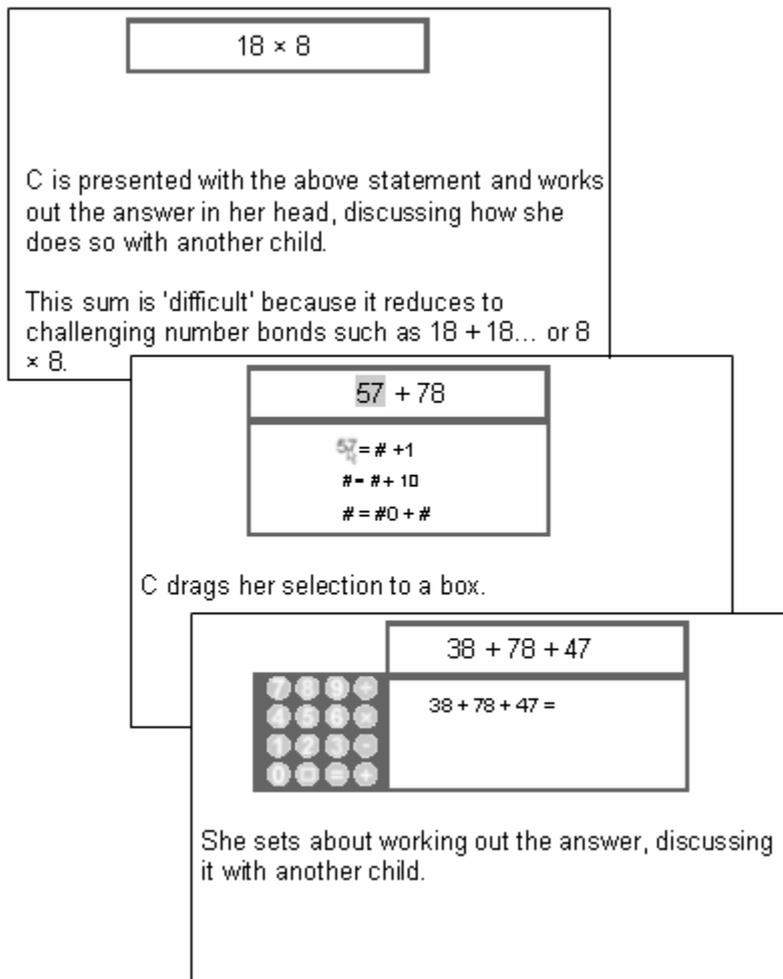
In the remainder of this section the evolutionary development of the task and method is traced out. Reflections on, and guidelines for, storyboarding are set out in the following section.

3.2.1 From abstraction to concretisation

As stated in Section 2.3, the switch from interface schematics to storyboards was first motivated by a need to capture the mathematical learning content of thought experiments without concern for technical implementation. To this end the interface was abstracted to a bare minimum and C (the epistemic learner) was envisioned working directly with raw notation.

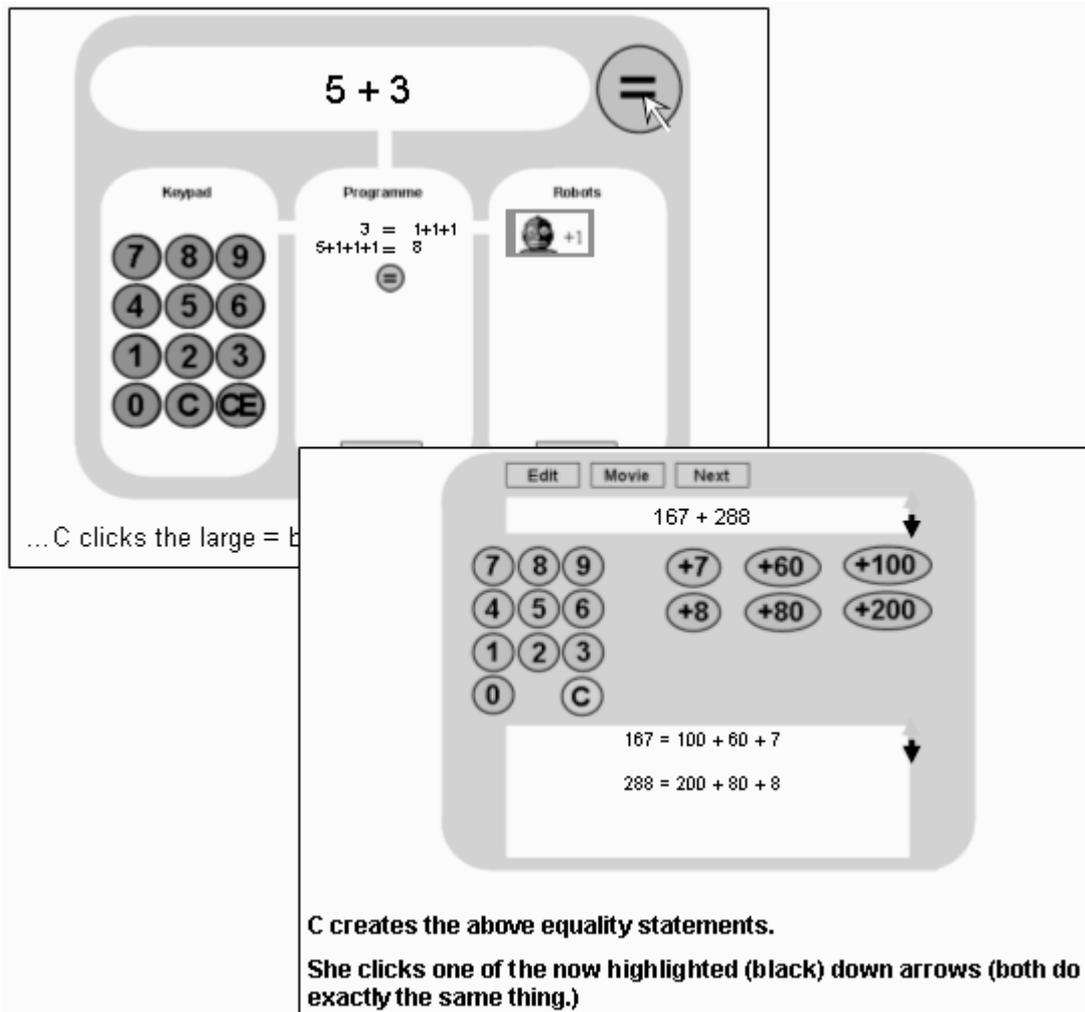
Figure 3a, the first slide of the first storyboard, shows an interface that has been abstracted to a single arithmetical statement. The following slides followed stepwise construction and transformation of notation towards the learning goal of programming a calculation strategy. A sense of a possible learning sequences emerged over the duration of VT1.ppt.

The interface became more fleshed out and increasing consideration was given to manipulation tools, such as the keypad shown in Figures 3b and 3c. Unlike the schematics produced previously, this storyboarded design had evolved incrementally over almost 100 slides that captured the learning content of thought experiments.



Figures 3a, 3b, 3c: Slides from the start, middle and end of VT1.ppt

When the second storyboard was created (VT2.ppt), the theme was continued but the interfaces became decreasingly abstract and began to look quite literal and concrete (Figure 4a). This learner-centred concretisation of the interface design continued through to the fifth storyboard, VT5.ppt (Figure 4b). This was no smooth journey. Many starts, stops and varied designs are apparent throughout the first five storyboards.



Figures 4a, 4b: Slides from VT2.ppt and VT5.ppt

3.2.2 Back to abstraction

However, the design ran into problems. These problems became apparent when the storyboards were shared with peers and presented at seminars. The interface design embodied the anticipated learning scenarios in a rather rigid manner and so lacked the flexibility to accommodate much variation. This rigidity arose from technological attempts to assist learners with the difficulties of translating ambiguous and assumptive natural language arithmetic into the explicit formal notation required by the task and goals. The tools and constraints did not so much assist with implementing learners' strategies as prescribe the designer's hoped-for strategies.

Explicit in the original schematics, and implicit in the early storyboards, was an assumption that the interface would be an evolution of the *Equivalence Calculator* from the previous small study (Section 2.3). Although this interface assumption was deliberately not represented in the early storyboards it reasserted itself gradually and appeared to interfere with the intended focus on learning. The designer had perhaps become too familiarised with the design to see this for himself. Peer feedback proved critical to the evolving thought experiments and was made possible because the storyboards provided sharable documentation.

With some reluctance a return was made to a stripped-down (abstracted) representation of the interface for the next storyboard (VT6.ppt). The literal

appearance of a black box marked “TOOLS” in a few subsequent slides served a somewhat quirky purpose: to remind the designer to keep it firmly shut for the meantime.

3.2.3 Participants and call outs

Another decision made in the light of peer feedback was the inclusion of an icon for C, and call outs (speech and thought bubbles). Along with the black box marked “TOOLS”, this helped keep the focus on C’s mathematical activity without a drift back to interface concretisation. It also provided a neat and instantly accessible way to present hypothesised “transcript excerpts” that might evidence mathematical thinking in real trials. While textual descriptions could have conveyed this information adequately, icons and call outs provided a visually intuitive format that aided peer engagement when sharing storyboards. The icon and call out took little effort to construct and were then replicated every time the current slide was copied and pasted to make the next.

It was envisioned that pilot trials would involve pupils using the computer-based task collaboratively with research guidance and encouragement (Pratt, 1998). In the subsequent storyboard (VT7.ppt) an icon of the participant researcher, R, was also included. The icon provided a natural way to capture R’s possible interventions and interactions with C as imagined in the thought experiments (Figure 5). The C and R icons were maintained throughout the storyboards (though are not shown in slides when not “speaking” or “thinking” to reduce clutter). This helped solidify the focus on collaborative learning without concern for interface implementation.

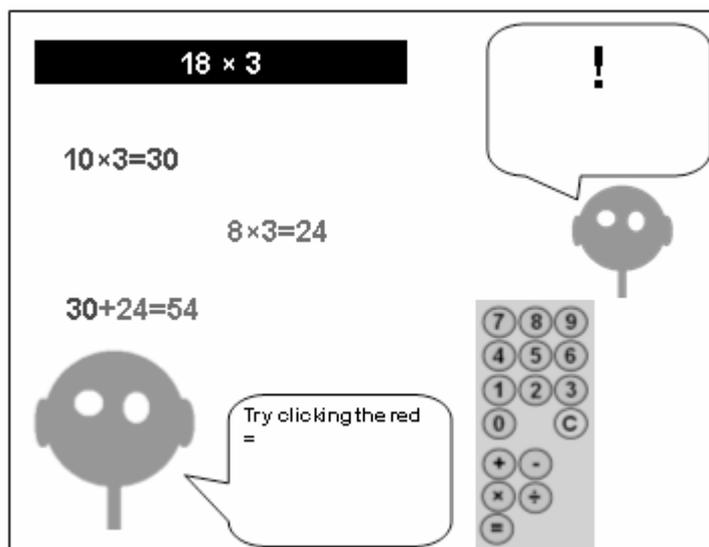


Figure 5: Participant icons and call outs

The storyboarding method was itself solidifying by this point. The initial need, to abstract interface design and focus on hypothesised learning sequences, had been reinforced by the repeated incremental emergence of concretisation. This was followed by the introduction of icons and call outs to help capture the designer’s assumptions regarding collaborative learning. In this sense, the storyboards had become a tool for capturing “virtual data”: analysable, sharable thought experiments about notational activity, pupil discourse and the role of R in imagined trials.

3.2.4 Importing real-world data

It was mentioned in section 3.2.2 that previous interface concretisations prescribed the designer's hoped-for-strategies. This issue was not adequately addressed by the return to abstraction or inclusion of participant icons and call outs. There was a need for real-world arithmetical strategies from children in classrooms. The QCA (2001) website *National Curriculum in Action* hosts samples of pupils' work with reference to levels of attainment as specified by the National Curriculum. This data enabled the importation of real pupils' arithmetical strategies for storyboard scenes. For example, Figure 6 shows a slide from VT9.ppt that illustrates "Tony's" written explanation for calculating $30+41$.

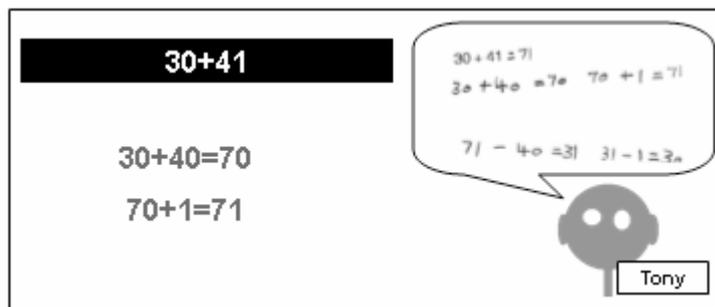


Figure 6: "Tony's" written strategy

Slides were created by imagining pupils inputting their own written strategies to explore the problems that might arise. These turned out to be the same problems that had emerged in the interface concretisations in early storyboards. When arithmetical strategies are spoken in natural language the partitioning and re-organising of notation is implicit. For example, "Tony" expresses composition explicitly using equalities such as $30+40=70$, but draws implicitly on arithmetical facts such as $41=40+1$. There appeared to be a need to give learners access to the arithmetical assumptions underlying their strategies in order to express them using the computer.

3.2.5 Towards implementation

Despite the remaining problems an implementable design had now been reached as reflected by the convergence across the final few storyboards. This became possible for two complementary reasons.

First, it had become apparent that the functionalities of the processing tools could be ascribed directly to the notation itself: notation could simply be clicked to make selections and transformations. Banishing all interface tools to a black box temporarily was helpful in discovering this. Two keypads were then needed for entering notation. The *Equivalence Calculator*, used in the small previous study, proved ideal without the need for much redesign (Figure 1).

Second, the problem of making implicit arithmetic explicit, that had plagued all ten storyboards, could simply be ignored in terms of technical implementation. The introduction of the C and R icons and call outs had enabled researcher interventions to be employed in storyboards. This was somewhat heavy-handed, involving exposition. However by drawing on the role of R the task could at least be implemented and piloted.

The software was programmed in Imagine Logo (Kalas & Blaho, 2003) which lends itself to the fast proto-typing of microworlds. Implementation occurred rapidly and little re-programming or debugging has proved necessary to date despite substantial trialing in a variety of primary schools (see Jones, 2007).

3.2.6 Towards piloting

A final significant change to the task design occurred during initial testing of the software by the designer. In the interests of speed, only addition operators had been included on the keypads at this stage. The storyboards were used as test plans and attempts to program the (addition) strategies from the *National Curriculum in Action* website were made. The impasse of inputting the partitions and commutations implicit in written strategies remained apparent and it was difficult to imagine offering learners an intuitive way to do this. However, upon programming some strategies, it was discovered the outcome made for an intriguing puzzle in which the goal is to use provided equalities to transform an arithmetical sum into its answer.

The impasse was finally overcome by switching things around: instead of children *building on* their existing explanations, they would attempt to solve *provided puzzles that appealed to* their existing explanations. From the designer's experimentations with solving puzzles, it seemed that this would give learners opportunities to make meaningful distinctions between compositional ($a+b=c$), commutative ($a+b=b+a$) and partitional ($c=a+b$) notation due to the distinct substitutive effect of each form. This would be an interesting finding in light of existing educational literature (Section 2.1).

A sequence of puzzles was created for trialing with pairs of Year 5 pupils in a cross-section of schools. The results from these trials seem promising (Jones, 2007). A screenshot of transcribed audiovisual data from subsequent trials of the prototype is shown in Figure 7 (the image was produced using *Transana* (Woods & Fassnacht, 2007)).

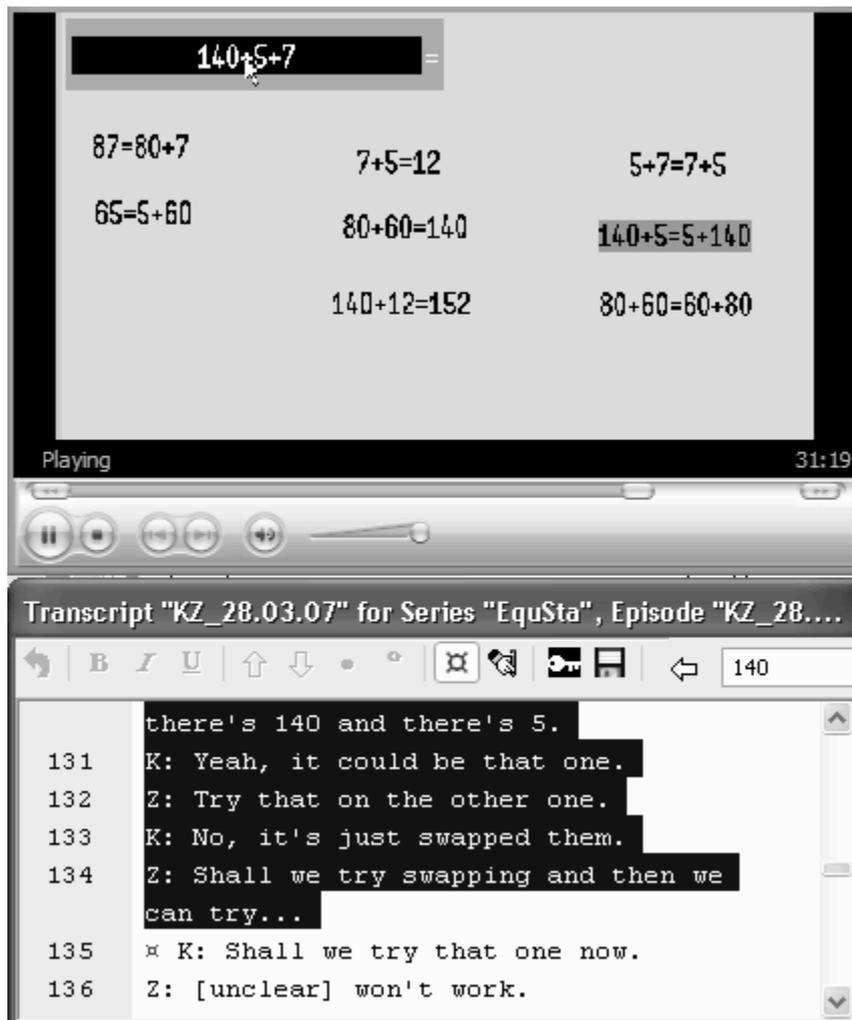


Figure 7: Screenshot from a trial of the task

4. Discussion

In this section the application of the storyboarding to the design of a mathematical microworld is reflected on in terms of its scope, strengths and pitfalls.

4.1 The role and scope of storyboarding

Storyboarding offers a method for developing initial rough ideas for task design. The need for it arose from an inability to get a working prototype off the ground through sketching interface schematics. As soon as the design seemed feasible (after the tenth storyboard), it was implemented and piloted.

The reasons a designer may struggle to implement an initial prototype are likely to be contextual. In this case the method was developed and used as part of a doctoral study and helped compensate for limited design experience. As is common with education doctoral studies, the research was carried out largely by an individual and the documenting and sharing of thought experiments helped compensate for a lack of daily discussion and feedback amongst research team members.

The use of storyboarding as a medium for designers, teachers and learners for the collaborative design of learning materials in classroom settings was not explored here, but offers promise (c.f. collaboration of designers and users in HCI storyboarding).

Conceivably, storyboards might also lend themselves as a medium of ideas for geographically disparate design teams.

4.2 Towards principles for storyboarding microworlds

Only one example of storyboarding has been presented here and no attempt at generalisation can be made. Furthermore the method was experimental in origin and developed as it was used. No initial principles were stated and adopted in order to do this. However, the method can be reflected on in terms of what elements would be maintained, and what would be avoided, were the exercise to be repeated in the bootstrapping of a new task design.

Elements to maintain would include...

- *Abstracted interfaces.* The functionality of inputting and manipulating arithmetical notation was put in a “black box”. The focus instead was on mathematical learning. Interestingly, the final task more closely resembles the abstracted slides than those where explicit attempts to flesh out the interface were made.
- *Participant icons and call outs.* Design is shaped by the assumptions and theoretical beliefs of the designer and these must be identified and made explicit (Pratt, Noss, Jones & Prodromu, submitted). The introduction of C and R icons and call outs allowed assumptions and hypotheses about collaborative mathematics learning to be captured. The design became feasible for implementation when the role of R was included as a central element of the task.
- *Imported classroom data.* Written arithmetical strategies from the *National Curriculum in Action* website were used. This allowed the designer to break out of using his own prescriptions of the strategies pupils might use.
- *Peer feedback.* The role of peer feedback is crucial in research and the storyboards opened a channel for sharing thought experiments. This is a key strength of the method.
- *Storyboards as test plans.* Storyboards provide an initial test plan for the first prototype of a microworld. This process led to a task design change: the switch from strategy programming to puzzle solving.

Elements to avoid would include...

- *Hidden assumptions.* It proved easier said than done to abstract interface assumptions away. Although not always concretely represented, an interface assumption grounded in the *Equivalence Calculator* design haunted the early storyboards. Conversely, assumptions about collaborative learning were in need of concretisation but this was not realised until VT7.ppt.
- *Proceduralism.* Overcoming an implementation impasse is no smooth, procedural process. It involves exposing hidden assumptions and discovering new vantage points on a hypothesised task. Post-hoc experimentation with the software prototype threw up the approach of allowing pupils to solve puzzles rather than construct strategies. It would have been possible to storyboard pupils manipulating notational constructions, rather than constructing notation themselves, had the idea occurred earlier.

- *Inefficiency.* A lot of slides were created and storyboarding continued for six months. During this time the method itself was refined and retrospectively guidelines have been identified. Hopefully these guidelines can enable greater efficiency and quicker results when using the method.

5. Conclusion

Storyboarding was developed in response to an implementation impasse when sketching initial interface schematics. The abstraction of interface details and capture of thought experiments involving pure arithmetical notation helped shift the design focus away from technological concerns and onto learning concerns. This shift needed to be reemphasised in the light of peer feedback that exposed rigidity and a return to techno-centricity over the duration of storyboards. This was addressed by representing the manipulation tools as a black box, introducing pupil and researcher icons and call outs, and importing pupils' written strategies from the classroom.

The storyboarding method offers designers a medium for expression, and the storyboards produced can be considered as virtual qualitative data that provides an audit trail for dissemination, feedback and accountability. This paper offers one such example of how storyboards can support the systematisation and dissemination of thought experiments.

References

- Barron, B. (2000). Problem solving in video-based microworlds: Collaborative and individual outcomes of high-achieving sixth-grade students. *Journal of Educational Psychology*, 92, 2, 391-398.
- Behr, M., Erlwanger, S., & Nichols, E. (1976). How children view equality sentences. *Technical Report PMDC-TR-3, 1-10*. Tallahassee: Florida State University.
- DfES (2002). Teaching of calculation in primary schools. *HMI Report 461*. [WWW document]. URL <http://www.ofsted.gov.uk/assets/2313.pdf>
- Hewitt, D. (2003). Some issues regarding formal algebraic notation. *23rd Conference of the British Society for Research into Learning Mathematics*, 1, 31-36.
- Jantke, K., & Knauf, R. (2005). Didactic design through storyboarding: standard concepts for standard tools. *First International Workshop on Dissemination of ELearning Technologies and Applications*, 20-25. Cape Town: South Africa.
- Jones, I. (2007). Arithmetical notating as a diagrammatic activity. *Proceedings of the British Society for Research into Learning Mathematics*, 27, 2, 31-36. Sheffield Hallam University.
- Jones, I. (2006). The design of equality statements. *Working Papers of the Warwick SUMINER Group*, 2, 63-82.
- Kalas, I., & Blaho, A. (2003). Exploring visible mathematics with imagine: Building new mathematical cultures with a powerful computational system. In G. Marshall & Y. Katz (Eds.), *Learning in School, Home and Community* (pp. 53-64). IFIP Conference Proceedings. Manchester: Kluwer.
- Knauf, R., Sakurai, Y., Dohi, S., Tsuruta, S. and Gonzalez, A. (2007). Toward making didactics a subject of knowledge engineering. *Seventh IEEE International Conference on Advanced Learning Technologies*, 788-792. Nigata: Japan.

- Knuth, E., Stephens, A., McNeil, N., & Alibali, M. (2006). Does understanding the equals sign matter? Evidence from solving equations. *Journal for Research in Mathematics Education*, 37(4), 297-312.
- Madsen, K. & Aiken, P. (1993). Experiences using cooperative interactive storyboard prototyping. *Communications of the ACM*, 36, 4, 57-64.
- Madsen, K. & Aiken, P. (1991). Cooperative interactive storyboard prototyping. In S. J. Andriole, (Ed.), *Storyboard Prototyping: A New Approach to User Requirements Analysis*, Second edition. Wellesley, MA: QED Information Sciences.
- Molina, M. (2006). *Desarrollo de pensamiento relacional y comprensión del signo igual por alumnos de tercero de educación primaria*. Unpublished Doctorate. Facultad de Ciencias de la Educación: Universidad de Granada.
- Papert, S. (2006). Afterword: After how comes what. In R. K. Sawyer (Ed.) *The Cambridge Handbook of the Learning Sciences* (pp. 581-586). New York: Cambridge University Press.
- Pratt, D. (1998). The construction of meanings in and for a stochastic domain of abstraction. Unpublished doctoral thesis. Institute of Education, University of London.
- Pratt, D., Noss, R., Jones, I. & Prodromu, T. (submitted). Designing for mathematical abstraction. *Journal of the Learning Sciences*.
- QCA, (2001). *Mental calculations – James and Tony*. [WWW document]. URL <http://www.ncaction.org.uk/items/pdf/593.pdf>
- Woods, D. & Fassnacht, C. (2007). *Transana v2.20*. URL <http://www.transana.org>. Madison, WI: The Board of Regents of the University of Wisconsin System.