Interaction Design for the Resolution of Linear Equations in a Multimodal Interface

Silvia Fajardo Flores and Dominique Archambault

EA 4004 – CHArt-THIM
Université Paris 8, France
silvia.fajardo-flores@etud.univ-paris8.fr,
dominique.archambault@univ-paris8.fr

Abstract. This article belongs to the field of Human-Computer Interaction, in the context of the access to Mathematics for people with visual disabilities. In a school scenario, the students with blindness who learn Algebra need to work on mathematical expressions, to collaborate and to communicate with their classmates and teacher. This interaction is not straightforward between students with and without sight, due to the different modalities they use in order to represent mathematical contents and to work with them. The computer presents a great opportunity to promote this type of interaction, because it allows the multimodal representation of mathematical contents. After the conduction of experiments on linear equation solving with students with and without sight, we have modelled their intentions and actions and we present a proposal for the interactions required in a multimodal interface serving this purpose. Lastly, we consider the possibilities and limitations for implementation.

Keywords: visual disability, accessibility, mathematics, HCI

1 Introduction

The access to mathematics and sciences in general for students with visual disabilities remains a challenge for both educators and researchers. The difficulties are diverse, and they concern the presentation and communication of contents, and the facilitation to perform calculations [1, 2]. In a school scenario, students who begin to learn Algebra need to take notes, understand and solve exercises, practise operations related to expressions and solving methods, and communicate with their teacher and peers. This interaction is complicated between students with and without sight, due to the different representation of contents they use. The computer presents a great opportunity to promote interaction between sighted and non-sighted people, because it allows for the multimodal representation of contents and the facilitation of simultaneous access in a common interface. The design of an accessible interface for linear equation solving requires a thorough understanding of the needs of sighted and non-sighted users. In a previous experiment we have identified the user intentions for linear equation solving [3]. The aim of this paper is to present the modelling of those actions, and to propose the interaction features to enable sighted and non-sighted users to work in a synchronised interface.

2 Software Support to Do Mathematics

There exist educational software allowing a complete range of manipulations to work with Algebra at least at a basic level. Examples of this type of software are APLUSIX [4], PIXIE [5] and VP Algebra. A different type of software to do mathematics is the Computer Algebra Systems (CAS), such as MuPAD, Sage, Axiom and Mathematica. These CAS function as symbolic calculators, receiving an expression as input and returning the requested answer with no feedback of the solving process, and therefore they are not recommended for students who begin to learn Algebra. Though they have been used for more advanced calculations, they are suggested to represent an obstacle for the development of algebraic symbolism in students [6, 7], due to the finding that in the context of learning with a computer-based environment, the interactions on the machine will play an important role in the student's construction of meaning [8, 9]. Following an analysis of the didactical problems related to the functionality of CAS, the Texas Instruments company developed the Symbolic Math Guide (SMG) software for its TI-92 calculator [10]. The SMG allows users to choose the transformations to be applied to the equations and gives feedback on the partial result. All these software applications, though complete in functionality, are not fully accessible for use with screen readers or Braille displays.

The accessibility of mathematical contents for learning has been the subject of several research and software development projects. The Math Genie [11, 12] was one of the first efforts to aid students with blindness understand the structure of expressions. Using the keyboard, students are able to browse equations, fold and unfold its subexpressions; the software uses both visual and audio output. The Lambda system [13] is the result of a European project aimed to facilitate the edition and manipulation of expressions for students with blindness. Lambda allows users to write mathematical expressions in a proprietary linear notation, and uses an 8-dot Braille code output in combination with a screen reader. The MaWEn prototypes have been developed for experimenting interaction models. They are based on the synchronous presentation of two views: graphical and Braille, supporting multiple codes; they allow trans-modal pointing and selection of terms or subexpressions [2, 14], and include assistants for simplification and manipulation [1, 15]. Even though these software and prototypes are accessible and were conceived to facilitate doing mathematics to students with blindness, their possibilities of manipulation are limited.

3 Linear Equation Solving: Understanding Student Goals and Actions

It is suggested that the solution of linear equations belongs to a domain that can be easily characterised and studied [16]. In a study with sighted participants conducted by [17], the solving strategies and actions performed by college students were analysed and organised in three stages:

Attraction Organisation of occurrences of the unknown in a way that they can be simplified further. e.g. $3x + 1 = x + 2 \rightsquigarrow 3x - x = 2 - 1$

Collection Addition of common terms in order to reduce the occurrences of the unknown. e.g. $3x - x = 2 - 1 \rightsquigarrow 2x = 1$

Isolation Elimination of the structure that surrounds the unknown, with the purpose of finding its value. e.g. $2x = 1 \rightsquigarrow x = 1/2$

The results of the study showed that students repeated one or more of these stages depending on their solving strategy, and that the number of repetitions depended on the proficiency of the participant. On the other hand, in our previous study [3] we observed the actions of students with and without sight using an oral protocol, with the aim to compare their needs and look for possible differences in intentions or actions. The exercises included in the experiment required simplifying common terms, multiplying monomials, binomials and polynomials, and solving a linear equation, which were identified in [18] as essential tasks in basic Algebra. It is suggested that equation solving demands the systematic execution of actions depending in the individual strategy, with no regard of visual ability. In terms of interactions, we identified two critical requirements: direct access to specific terms of the equation and minimisation of the user's mental load. These features are considered as fundamental in our proposal.

4 Action Modelling

The actions performed by the participants of our previous study can be categorised in stages, some of which can be matched with those from the study by [17]. These stages represent the intentions of the participants, expressed either explicitly or implicitly.

4.1 Verification

Consists of the analysis of the state of the equation throughout the solving process, beginning by grasping the structure of the equation and the preparation of the solving strategy, continuing by checking the partial result of the applied operation, or searching for other information.

4.2 Simplification: Attraction and Collection

Simplification is the most basic and frequent stage. In our context, it consists in organising and adding common terms. In the frame of the analysis by [17], simplification consists of a combination of the stages of *Attraction* and *Collection*. We will use these terms in our discussion as means to characterise systematically the actions required to carry out a simplification.

4.3 Distribution

Solving a linear equation often requires multiplying monomials, binomials and polynomials. Performing these operations without an adequate external support puts into evidence the limitations of human memory, such as the difficulty of remembering multiple terms and the consequent need to look for them constantly.

4.4 Isolation

This stage consists of removing the surrounding structure of the unknown. It requires the application of transformations on both sides of the equation, which operationally could also be achieved by transposing a term to the opposite member and changing its sign. *Isolation* shares some common actions with *Attraction*.

Table 1 shows the different stages in resolution carried on by two participants. It can be observed that the solving strategies vary, but the actions are similar.

Table 1. Solving strategies of two participants.

Equation	Stages/Actions
$x+2(x+2(x+2))=x+2$	**Attraction** Transpose x, $+2$ Change sign
$-x-2+x+2(x+2(x+2))=0$	**Distribution** Multiply $2(x+2)$
$-x-2+x+2(x+2x+4)=0$	**Collection** Add common terms (in parentheses)
$-x-2+x+2(3x+4)=0$	**Distribution** Multiply $2(3x+4)$
$-x-2+x+6x+8=0$	**Verification** Find common terms in x
	Collection Add common terms
$-2+6x+8=0$	**Verification** Find independent terms
	Arithmetic simplification Add common terms
$6x+6=0$	**Isolation** Transpose 6 Change sign
$6x=-6$	Divide by 6
$x=-1$	

Equation	Stages/Actions
$x+2(x+2(x+2))=x+2$	**Attraction** Add $-x$ on both sides
$-x+x+2(x+2(x+2))=x+2-x$	**Collection** Eliminate instances of x
$2(x+2(x+2))=2$	**Isolation** Divide by 2
$x+2(x+2)=1$	**Attraction** Transpose x Change sign
$2(x+2)=1-x$	**Distribution** Multiply $2(x+2)$
$2x+4=1-x$	**Attraction** Add x on both sides
$x+2x+4=1-x+x$	**Collection** Add common terms (left member)
$3x+4=1-x+x$	Add common terms (right member)
$3x+4=1$	**Isolation** Add -4 on both sides
$-4+3x+4=1-4$	**Arithmetic simplification** Add common terms
$3x=-3$	**Isolation** Divide by 3
$x=-1$	

5 Interaction Design Proposal

The main objectives of the interface we aim to design are the facilitation of the manipulation tasks involved in learning Algebra, and the communication of contents between sighted and non-sighted students. It is important to clarify that we are not implying that learning Algebra is about manipulating equations. The teaching method must be determined by the teacher, while our interface will enable students to write equations, work with them, and communicate with peers.

The features proposed here aim at facilite access while maintaining automatisation features to a minimum, so that it is the user and not the system that produces the results. Some requirements from other works are implicitly taken into account, such as: navigation within the expression [1, 2, 11], minimisation of errors caused by memory limitations, and localisation of relevant positions [15]. On the other hand, it is important to consider the allowance for error making, which is inherent to the learning process. In this regard, our proposal allows the possibility to commit execution and arithmetic errors. e.g., Errors resulting from mentally adding coefficients, or from applying an invalid operation such as adding terms of different variables or exponents.

5.1 Use Case Diagrams

The user actions and stages have been organised in use case diagrams. These diagrams, shown in Figure 1, include edition features such as type, delete, copy/paste and select/unselect terms. The use cases that require previous selection are indicated with an asterisk (*).

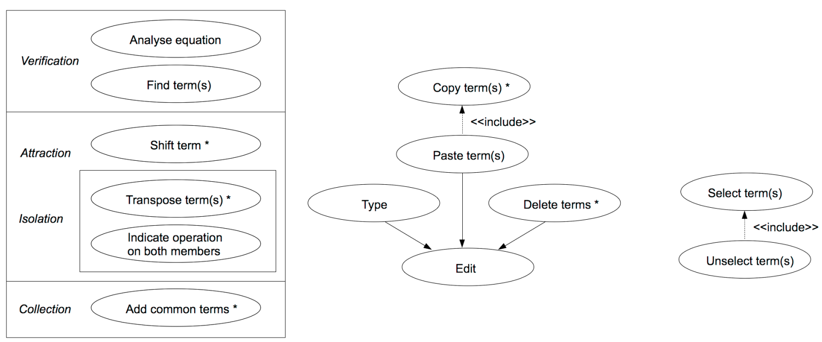

Fig. 1. Use Case Diagrams.

5.2 Interaction Features

The use cases can be put into operation by enabling typing, browsing and edition of equations in the interface. Edition involves the use cases for writing and transforming the equation in the stages of *Attraction*, *Collection* and *Isolation*. Browsing corresponds to the *Analyse equation* use case in the non visual modality, and enables the search for terms needed in most use cases. In addition to the manual work with the equation, our proposal includes auxiliary interactions intended to provide an option to minimise the number of key presses, accelerating the access while maintaining the nature of the task. We do not make a distinction between interactions for sighted and non-sighted users, but the options we propose aim to facilitate access for both. The interactions for each use case are organised and described as follows:

Edition Users will be able to type, delete and modify terms. Typing will involve the use of numbers, basic operators [+, -, *, /], the usual letters to represent variables, avoiding those that could result ambiguous such as f and d. Up and Down arrows will be used to indicate the beginning and the end of an exponent. Automatic conversion to Presentation MathML will follow each typed character, and will have in turn an audio output. Copy, Paste and Delete commands are considered in edition. Selection is required for many of the use cases: pressing a function key when the cursor is in the position of the desired term will select it. The selection of terms will be kept until the user explicitly indicates end of selection.

Browsing Equations will be shown on the screen and will also be available in audio on keyboard browsing. Visual and audio output and feedback will be synchronised. Users will be able to navigate: 1) by term or operator, 2) by element within the active term (coefficient, variable, exponent), and c) by line. The unity of navigation by default will be the term. In order to facilitate navigation between lines, we propose to keep the last position of the cursor in each line. Since selection will be persistent, we consider the possibility to alternate the cursor position from the active line to the list of selected terms. This could help minimise the change between lines and probably facilitate the follow-up on distributions.

Auxiliary Interactions Auxiliary interactions will be available through the use of commands. They include options for finding, writing or reorganising terms faster than if done manually. The proposed auxiliary interactions are: *Find Term [...], Find Common Terms in [...], Transpose, Add term to both sides, Divide by term,* and *Add coefficients.* e.g. A simplification would involve browsing the equation in order to find all terms of a certain exponent, for example x^2, then adding them and writing their sum in the resulting line; by using commands, the operation could be done faster by using either or both commands *Find Common Terms in* $[x^2]$ and *Add coefficients.*

5.3 Implementation Issues and Limitations

The freedom of browsing, writing and manipulation, as well as the allowance for error considered in our proposal, require a high degree of difficulty in implementation. Contrary to literary text, mathematical expressions are bi-dimensional; the current state of technology to display them involves the use of mark-up languages such as Presentation MathML, Content MathML and OpenMath, which allow to display the expressions but do not allow direct browsing and edition. For the implementation of this proposal we have chosen Presentation MathML, because it has a more direct correspondence with human reading and writing than Content MathML, and therefore it allows immediate translation. In addition to that, other software tools have been developed to manipulate Presentation MathML. In order to provide audio feedback, we have considered two possibilities: 1) synchronisation with the user's screen reader, or 2) use of an internal screen reader.

While users are accustomed to browsing and editing by character, which is the default behaviour for text editors, the introduction of default browsing behaviour by term or by subexpression might affect the user's perception of the consequent edition behaviour. Regarding deletion of terms, we have considered a delete-within-current-term protocol as default, and an alternative select-first-then-delete protocol. The replacement of existing terms could imply the introduction of an explicit indication that a change is going to be made to them, in order to avoid errors of unwanted replacement. On the other hand, the introduction of a replacement mode would add complexity for the user.

The feature of persisting selection is particularly subject to observation, since its behaviour differs from text edition where selection is lost on change of cursor position. In the case that this feature would be confusing for the user, an alternative option for marking terms is considered, in order to differentiate it from common selection. All features proposed here are subject to user testing, in order to find out the protocols that suit best each interaction.

5.4 Conclusion

The features desired in a multimodal interface to solve linear equations and the operations related to them could be summarised in: facilitation of direct access to the terms of the equation and minimisation of mental load. After analysis and categorisation of the user actions, we have defined the interactions for the interface, taking into account both visual and non-visual representation. The proposal presented here considers browsing by term and subexpression, which might affect the users' perception on the behaviour of the edition features in relation to text editors. Having considered several alternatives of interaction, we expect this proposal to serve as basis for the development of a prototype which will be subject to user testing, in order to validate its effectiveness from the perspective of students and teachers. By providing a way to facilitate the transformation of equations, and along with an adequate teaching method, this prototype is expected to be useful to facilitate the understanding of algebraic concepts, since it will be the user and not the system who will decide on the transformations and produce the results.

References

1. K. Miesenberger. Doing mathematics. http://www.ascience-thematic.net/en/conferences/paris/Miesenberger, 2008.
2. D. Archambault. Non visual access to mathematical contents: State of the art and prospective. In *Proceedings of the WEIMS Conference 2009 (The Workshop on E-Inclusion in Mathematics and Science)*, pages 43–52, 2009.
3. S. Fajardo Flores and D. Archambault. Understanding algebraic manipulation: Analysis of the actions of sighted and non-sighted students. In K. Yamaguchi and M. Suzuki, editors, *The International Workshop on Digitization and E-Inclusion in Mathematics and Science 2012 (DEIMS12)*, 2012.
4. J. F. Nicaud. A general model of algebraic problem solving for the design of interactive learning environments. In J. et al Ponte, editor, *Mathematical Problem*

Solving and New Information Technologies, volume 89 of *NATO ASI Series*, pages 267–285. Springer Berlin Heidelberg, 1992.
5. D. H. Sleeman. Inferring (mal) rules from pupils' protocols. In *Proceedings of the European Conference on Artificial Intelligence*, pages 160–164, 1982.
6. M. Artigue. Learning mathematics in a cas environment: The genesis of a reflection about instrumentation and the dialectics between technical and conceptual work. *International Journal of Computers for Mathematical Learning*, 7:245–274, 2002.
7. J. Monaghan. Computer algebra, instrumentation and the anthropological approach. *International Journal for Technology in Mathematics Education*, 14(2):63–72, 2007.
8. N. Balacheff. Artificial intelligence and mathematics education: Expectations and questions. In T. Herrington, editor, *Proceedings of the 14th Biennal of the AAMT*, pages 1–24, 1993.
9. P. Drijvers and K. Gravemeijer. Computer algebra as an instrument: Examples of algebraic schemes. In D. Guin, K. Ruthven, and L. Trouche, editors, *The Didactical Challenge of Symbolic Calculators: Turning a Computational Device Into a Mathematical Instrument*. Springer, 2005.
10. A. Özgün Koca and M. T. Edwards. Symbolic math guide: An innovative way of teaching and learning algebra using ti-89 and ti-92+ graphing calculators. In *Proceedings of the 2nd International Conference on the Teaching of Mathematics (at the undergraduate level)*. John Wiley & Sons Inc., 2002.
11. D. Gillan, P. Barraza, A. Karshmer, and S. Pazuchanics. Cognitive analysis of equation reading: Application to the development of the math genie. In K. Miesenberger, J. Klaus, W. Zagler, and D. Burger, editors, *Computers Helping People with Special Needs*, pages 630–637. Berlin: Springer-Verlag, 2004.
12. A. Karshmer, C. Bledsoe, and P. Stanley. The architecture of a comprehensive equation browser for the print impaired. In K. Miesenberger, J. Klaus, W. Zagler, and D. Burger, editors, *Computers Helping People with Special Needs*, pages 614–619. Berlin: Springer-Verlag, 2004.
13. W. Schweikhardt, C. Bernareggi, N. Jessel, B. Encelle, and M. Gut. Lambda: a european system to access mathematics with braille and audio synthesis. In K. Miesenberger, J. Klaus, W. Zagler, and A. Karshmer, editors, *Proc. ICCHP 2006 (10th International Conference on Computers Helping People with Special Needs)*, pages 1223–1230. Springer, 2006.
14. D. Archambault, B. Stöger, M. Batusic, C. Fahrengruber, and K. Miesenberger. A software model to support collaborative mathematical work between braille and sighted users. In *Ninth International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 07)*, 2007.
15. B. Stöger, K. Miesenberger, and M. Batusic. Mathematical working environment for the blind: Motivation and basic ideas. In K. Miesenberger, J. Klaus, W. Zagler, and D. Burger, editors, *Computers Helping People with Special Needs*, pages 656–663. Berlin: Springer-Verlag, 2004.
16. J. Anderson. Human symbol manipulation within an integrated cognitive architecture. *Cognitive Science*, 29:313–341, 2005.
17. L. Carry, C. Lewis, and J. Bernard. Psychology of equation solving: An information processing study. Technical report, The University of Texas at Austin, 1979.
18. B. Stöger, M. Batusic, K. Miesenberger, and P. Haindl. Supporting blind students in navigation and manipulation of mathematical expressions: Basic requirements and strategies. In Computers Helping People with Special Needs, editor, *Miesenberger, K. and Klaus, J. and Zagler, W. and Karshmer, A.*, pages 1235–1242. Berlin: Springer-Verlag, 2006.