Manipulation of Mathematical Expressions in Collaborative Environments

Marco Pollanen[1(✉)], Sohee Kang[2], and Bruce Cater[1]

[1] Trent University, Peterborough, Canada
`{marcopollanen,bcater}@trentu.ca`
[2] University of Toronto Scarborough, Toronto, Canada
`soheekang@utsc.utoronto.ca`

Abstract. Recent decades have seen phenomenal growth in the use of communication and collaborative technologies in many academic disciplines. There has, however, been little adoption of tools for online collaboration in post-secondary mathematics education. In this paper, we argue both that this may be due to limitations of mathematical interfaces and that the adoption of collaborative tools may provide significant pedagogical benefits. To date, mathematical user-interface research has focused primarily on mathematical expression input, and mostly from a perceptive of document creation or computer algebra system use by expert users. Little work has been done on the specific needs of novice users, including students, and even less work has considered the manipulation of mathematical expressions. In this paper, we outline some user-interface challenges of current input systems with respect to entry and manipulation of mathematical expressions by novice users, and we introduce a model that makes entry and manipulation easier for those users.

Keywords: Mathematical collaboration · Novice user interfaces · Mathematical formula input · Mathematical software · Post-secondary mathematics education

1 Introduction

Most Internet communication tools are text-based messaging applications that also allow for the transmission of useful videos, photos, and voice notes. Perhaps 25% of academic subjects, however, rely heavily on symbolic and diagrammatic content for knowledge transference – content that cannot be conveyed electronically in a form that allows for the rich interaction that occurs, say, between classmates or between a student and a professor at a blackboard during an office hour. This puts online students of these subjects at a particular disadvantage, relative to their counterparts in a traditional learning environment.

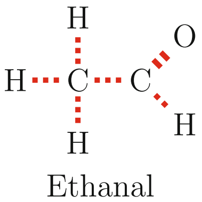
Ideally, the level of interaction between students and instructors should be based on what is pedagogically best. Consider a student in an online chemistry course, who may wish to seek clarification from their instructor as to the structure of a particular chemical (see Fig. 1). The structure should be communicated

P. Zaphiris and A. Ioannou (Eds.): LCT 2017, Part II, LNCS 10296, pp. 60–70, 2017.
DOI: 10.1007/978-3-319-58515-4_6

in a way that is easy to manipulate in order to demonstrate a process. A Psychology student may similarly wish to collaborate online with other students on a lab project, and to be able to communicate and manipulate a basic statistical formula. Moreover, a large number of studies, largely based on face-to-face classroom applications, extol the benefits of new interactive pedagogical models, such as peer-based learning [6]. Others have shown that out-of-class student-teacher interaction (e.g., office hour attendance) leads to improvements in many key academic measures, including student performance, retention, and satisfaction, while students themselves overwhelmingly show an interest in greater online communication with their professor [2].

Interaction and communication in courses rich in quantitative content is particularly important, as evidenced by the significant supporting resources universities allocate to such courses. Quantitative service courses, for example, are typically assigned a disproportionate level of tutorial and teaching assistant resources. There are usually counselling and academic skills programs in place to deal with general numeracy skills, and there are almost always mathematics and statistics tutorial centres available to deal with specific course content.

Rarely, however, are these supports replicated for online learners. Unfortunately, the level of interaction between students and instructors in online symbolic and diagrammatic-rich courses is more constrained by what is technologically possible, than it is based on what is pedagogically best. Intuitive technologies for interacting with symbolic and diagrammatic course content are not currently available, and given the online paradigm shift to mobile devices, it has only become more difficult to create symbolically rich content due to the interface limitations of smartphones and tablets. As a result of these technological limitations, there has been very little adoption of online communication tools in the quantitative sciences [1].



$$r_{xy} = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2}}$$

Fig. 1. Simple molecular structure (left) and an elementary statistical formula (right)

New tools that allow for non-text-based communication may lead to new pedagogical approaches that would be of particular value to online students. These tools may also lead to the ability to better engage students who are underrepresented in the academic discourse in quantitative courses. It has, for example, long been known that women are less likely to engage in classroom dialogue than their male peers in post-secondary mathematics [4]. Moreover, because English language learners, such as recently-arrived immigrants and international students, may be shy to engage in classroom conversation, new communication tools for the technologically-enhanced class may create a more inclusive student-centred environment leading to further democratization of learning.

1.1 Online Tools for Communication of Symbolic Academic Content

Teaching introductory quantitative service courses to a wide range of students presents a number of challenges and opportunities. In a first-year statistics course, for example, although some students will be statistics majors, the majority may be seeking only a single statistics requirement and may suffer from some "math anxiety". One symptom of this anxiety, which affects up to 85% of students [9], is that it prevents students from visiting their instructor's office hours, thereby undermining their rates of student success, retention, and satisfaction [7]. The anxiety may also represent a barrier for specific groups, including women and English language learners.

In [3], we explored the use of anonymity in online communication and found it to have dramatically improved participation rates for office hours from less than 10% of students in the class attending a traditional office hour, to over 80% attending via online delivery. Despite the potential of anonymity, our experience is that students in completely online courses have particularly low-levels of help-seeking behaviour due in large part to technological limitations.

While other areas of Internet communication have evolved at an astonishing pace, mathematical collaboration online remains a formidable challenge [12]. There are at least two reasons for this. First, a given piece of hardware must somehow allow for the inputting of hundreds of mathematical symbols. Second, the inherently two-dimensional structure of mathematical notation (see Fig. 2B) requires that spatial relationships between those symbols be accurately conveyed. There are, of course, current standards that allow for the text-based entry of mathematical expressions. But those standards have a steep learning curve and low human readability. The predominant standard for mathematical writing, TeX could be used to express, for example, the simple expression in Fig. 1 – a standard equation in any first-year Statistics service course. But in a live conversation, how realistic is it to expect first-year students to write the TeX representation of that equation, shown in Fig. 2C? Namely,

```
r_{xy} = \frac{n \sum_{i=1}^{n} x_{i} y_{i} - \sum_{i=1}^{n} x_{i}
\sum_{i=1}^{n} y_{i} }{ \sqrt{n \sum_{i=1}^{n} x_{i} ^{2} - \left(
\sum_{i=1}^{n} x_{i} \right)^{2} }  \sqrt{n \sum_{i=1}^{n}
y_{i}^{2} - \left( \sum_{i=1}^{n} y_{i} \right)^{2} } }
```

The main alternative to text-based input is a structure-based editor, such as that which is found in Microsoft Word. In such an editor, the user inserts individual symbols and mathematical structures, separately selecting them by clicking through tabs of buttons (see Fig. 2A). As with text editors, however, structure-based editors suffer from severe usability problems [8]. For example, in an observational study [1], we argued that structure-based editors usually force a user to write a formula in a manner that is different from how he or she would write it out by hand. To see this, consider the expression $\sqrt{x}/y$. The default behaviour of a structure-based editor forces the user to input the fraction first, followed by the root, and then the x and y. Intuitively, though, a person writing this expression using a pen and paper would likely write the root of x first, followed by the fraction bar, and then the y. In essence, then, the user of a structure-based editor must use an unintuitive order to input the expression, requiring the user to have the ability to mentally parse the mathematical expression into valid sub-expressions. This too may be an unrealistic expectation for students and other novice users who are struggling to understand complex expressions.

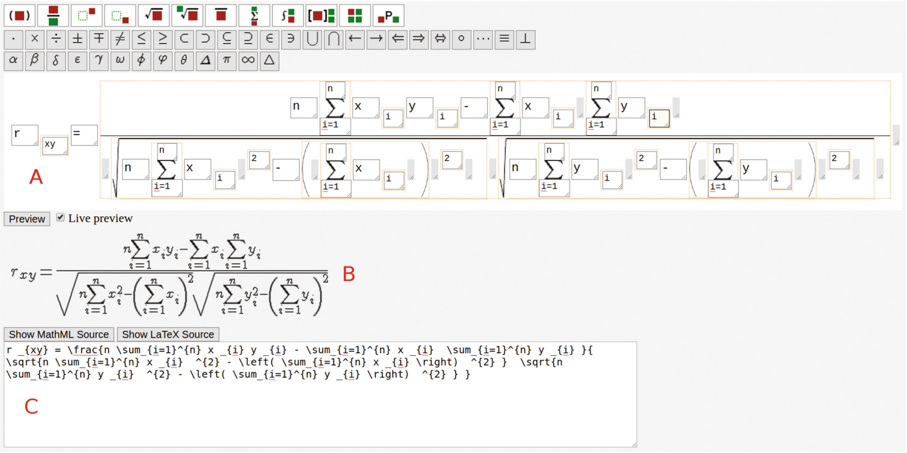
A completely different approach is handwriting recognition of mathematics, which uses the intuitive pen and paper paradigm for input. But that approach too has many limitations. Pen-computing hardware is still far from ubiquitous, and robust recognition of mathematical writing is still a formidable task [13]. Moreover, this paradigm is limited to the intuitive input of mathematics, and does not create a form that can be edited or modified [5].

A further and fundamental problem with all mathematical input systems to date is that they have been designed largely for document creation or interaction with a computer algebra software system, and not for collaboration.

We argue that real-time collaboration and communication has unique interface requirements. An instructor, while chatting online with a student, might, for example, want to ask the student to simplify the expression $\sqrt{\sqrt[3]{\left(\frac{x^2+x}{x}\right)^3}}$ to $\sqrt{x+1}$ in a step-by-step fashion. With any one of the traditional formula editor models, it is very difficult to accomplish this type of rich interaction, for they were designed only to create expressions, and it is often easier to create a new formula from scratch than it is modify an existing formula. To overcome this problem, we introduce a user interface model that, much like the pen-based system, is based on drawing a representation of the expression, but that is also based on a diagram-editor UI model in which symbols are selected from palettes (or other shortcuts). And again like a pen-based approach, this follows a well-known UI model, so there is little learning curve, but expression recognition rates are dramatically better and the model is consistent with most hardware interfaces from keyboard/mouse to touch-based interactions. Moreover, modifications of expressions can occur under the same UI model, unlike pen-based systems where input and modifications occur under different interaction models. Usability results suggest that not only that this allows for faster and more intuitive input from novice users, but also that mathematical expressions can be easily interacted with and modified, making it well-suited for collaborative environments.

2 Input and Manipulation of Mathematical Expressions

There are a variety of mathematical input methods: handwriting-based, palette-based or text-based. In addition, these methods use various combinations of



$$r_{xy} = \frac{n \sum_{i=1}^{n} x_{i} y_{i} - \sum_{i=1}^{n} x_{i} \sum_{i=1}^{n} y_{i}}{\sqrt{n \sum_{i=1}^{n} x_{i}^{2} - \left(\sum_{i=1}^{n} x_{i} \right)^{2}} \sqrt{n \sum_{i=1}^{n} y_{i}^{2} - \left(\sum_{i=1}^{n} y_{i} \right)^{2}}}$$

```
r _{xy} = \frac{n \sum_{i=1}^{n} x _{i} y _{i} - \sum_{i=1}^{n} x _{i}  \sum_{i=1}^{n} y _{i} }{
\sqrt{n \sum_{i=1}^{n} x _{i}  ^{2} - \left( \sum_{i=1}^{n} x _{i} \right)  ^{2} }  \sqrt{n
\sum_{i=1}^{n} y _{i}  ^{2} - \left( \sum_{i=1}^{n} y _{i} \right)  ^{2} } }
```

Fig. 2. A common elementary statistics formula (B) along with its representation in a structure-based editor (A) and in TEX code (C).

stylus, keyboard, mouse or touch input over a variety of hardware form factors – computers, tablets, and smartphones. Despite a plethora of different mathematical input technologies, computer input of mathematics remains slow and cumbersome, relative to handwriting mathematics on a chalkboard. So, handwritten mathematics is thought of as the *gold-standard* for mathematical input.

However, mathematics written on a chalkboard is only a visual representation of mathematics. There is a dichotomy between the input of mathematics for visual presentation of mathematics, such as produced by software packages such as TEX, and the input of mathematics for semantic purposes, through the use of software packages such as Mathematica.

From a presentation standpoint, it is quick and easy for mathematics to be written on a chalkboard and for corrections, such as changing a plus to a minus sign, to be made. Great potential does, however, lie in striving to go beyond this model. Handwritten mathematics cannot be easily reused, searched, and edited. For example, when working on a step-by-step calculation, it might be more efficient to copy a line and edit it than to rewrite the expression each time. Likewise, an instructor conducting virtual office hours might gain efficiency by being able to search through old questions and reusing parts of explanations and expressions from common questions.

From a semantic standpoint, there is potential to create user interfaces that go far beyond just being able to replicate the efficiency of handwritten mathematics. Mathematical communication has always relied on a facilitating medium, such as a chalkboard. Even mathematical thought requires one to work out ideas on paper. However, if one were able to write mathematics that is semantically understood by a machine, perhaps an interactive interface can be designed to handle routine calculations though the use of a computer algebra system.

For example, something simple like grabbing a sub-expression and moving it from one side of an equation to another might result in the sign being automatically changed. By reducing the cognitive load associated with more mundane elements of calculation, it might be possible to have a writing environment that frees the mind, allowing one to think more deeply about core concepts.

Since there has not been much focus in the literature on the ability to edit and manipulate mathematical expression, we will briefly review palette-based and pen-based input systems from the perspective of entry and manipulation by a novice user – say, a student. From a user interface point of view, text-based entry interfaces for mathematics mimic that of text-editors, however, as pointed out before they require advanced knowledge to input mathematics properly and are inappropriate for novice users and so we will not discuss them further.

2.1 Structure-Based Editors

Structure-based models typically allow users to select structures from palettes and separately populate them with symbols. Structure-based editors make it easy to find symbols and structures, and guarantee well-formed expressions that could make it easier for inputting semantically into a computer algebra system. From a usability perspective, however, they tend to suffer from a number of difficulties and due to these there is recognition that these types of editors have a reputation of being unattractive to both inexperienced and advanced users [8].

One problem is that entry of an expression typically requires users to navigate menus of symbols and templates as well as enter characters, causing the user to frequently switch between the keyboard and mouse.

A much larger problem is that structure-based editors take the two-dimensional visual representation of an equation and represent it as a series of nested structures, usually represented by nested boxes (see Fig. 2A). To interact with, and navigate though, this structure requires the use of interface interactions that a novice user might not be familiar with. In [8], it was shown that there was a lack of consistency between editors and that navigation can defy WYSIWYG principles. Pressing the cursor key, for example, may cause the cursor to jump from one structure to the next in an unpredictable way, as the two-dimensional structure is navigated.

[1] has shown that novice users, who don't have experience with a structure-based editor, have greater difficulty in inputting mathematics, and often get stuck and cannot even complete their expression, let alone manipulate it. It was also shown that users are forced to write expressions in a different order than they would on a piece of paper, forcing them to mentally parse the expression. This could be particularly challenging for students who don't have a great deal of mathematical training.

2.2 Pen-Based Input

The allure of pen-based input is that it is natural, effectively mimicking the experience of using a piece of paper, allowing input with little effort, and requiring no ability to mentally parse an expression beyond that required to writing the expression on paper [14].

The first obstacle encountered is that pen-based systems are not yet commonly used. And even if they were widely used, a second obstacle is that pen-based input in its most basic use just creates a digital image of what is written onto the screen, making it no different from writing on paper. To move beyond this point, we must be able to recognize the handwritten mathematics.

Robust recognition of handwritten mathematics is still a challenge. It is typically done in two phases, the recognition of symbols and the recognition of structures. Recognizing symbols, alone, is a difficult task. With handwritten recognition of text clues, such as dictionary matches, or in cases of languages such as Chinese, where there are a great many characters, stroke order can give many clues. However, in mathematics many symbols are very similar (e.g., $\cdot, 0, O, o, \bullet, {}^{\circ}, \circ, \odot, \oplus, \otimes, \oslash, \emptyset, \phi, \ominus, \theta, \varTheta, \cdots$). And even once the symbols have been correctly identified, knowing precisely where the user intended to put those symbols makes structural identification difficult. For example, $4 \cdot 5$ and 4.5 could be hard to differentiate.

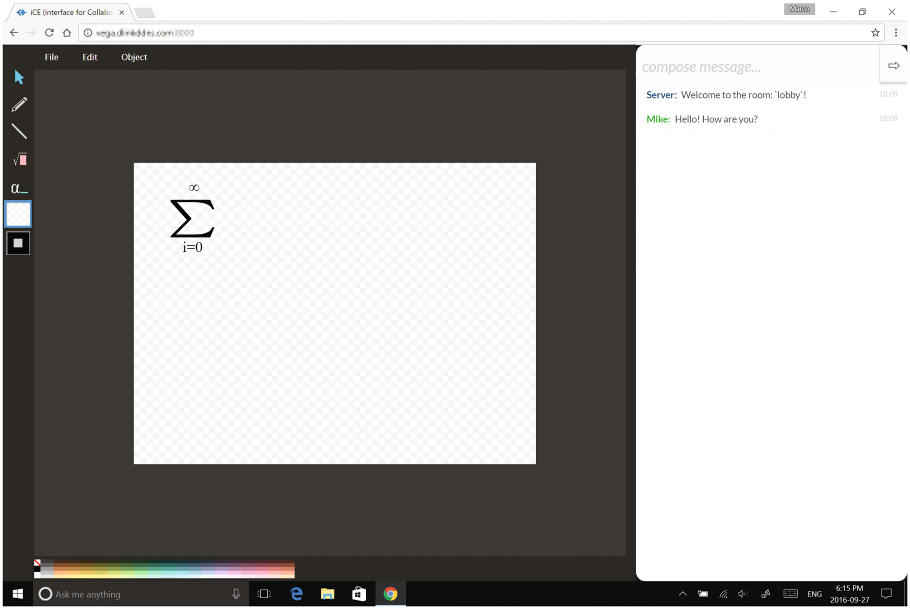
So, while writing mathematics by pen is an intuitive and effortless task, even for novice users, corrections will inevitably need to be made for mis-recognized input. This necessarily requires a change in interface model. For example, do you delete things with a lasso or by crossing out? Do you select the correct symbol from a pop-up menu of symbols? Such switching of user interface models can interfere with completing the task. Furthermore, is writing converted to typeset text as it is written, or are only full expressions? In either case, the user might find the sudden transformation jarring. Editing of already entered expressions would face the same sorts of challenges, but on a larger scale.

In the next section, we discuss a user interface model that is a hybrid of the palette and pen-based models that we have shown to be more intuitive for novice users and that has the potential to make the editing and manipulation of expressions more natural.

3 An Alternative Hybrid Input Method

As an alternative to the restrictive input approach of a structure-based editor and pen-based input, we proposed a collaborative environment based on a diagram editor user-interface model [13] – an open-source Web-application called iCE: interface for Collaborative Equations (see Fig. 3). Further in [12], we argued that this model is consistent with smartphone and tablet touch-based user-interface principles, and a mobile version of iCE was subsequently developed (see Fig. 4).

The advantage of the diagram-editor model for novice users is its familiarity to most users who have used office software that allows for the drawing of vector-based diagrams. In the case of mathematics, however, instead of just including resizable diagrammatic elements, such as lines and rectangles, the diagrammatic elements include re-sizable mathematical symbols, such as summation signs and brackets.



Server: Welcome to the room: `lobby`! 18:09

Mike: Hello! How are you? 18:09

Fig. 3. Screenshot of the desktop version of the iCE: interface for Collaborative Equations. On the left side is a shared workspace that allows for mathematical writing, and the right side is a chat window for conversation.

This approach is a hybrid of a palette-based editor and pen-based input – the user is able to select symbols from palettes, but is free to place the symbols anywhere they want on a canvas in order to create a 'picture' of their expression in a similar way to a pen-based system. One advantage over the pen-based system is that there is a subtle snapping of symbols to baselines, and so a baseline structural analysis algorithm [15] can be used to identify the expression. Unlike with handwriting, where the recognition failure rate is high, this approach allows for even very complicated mathematical expressions to be easily recognized [10]. The success is since, unlike handwriting analysis, the symbol was chosen from a palette it is known with certainty as well as is the location the user intended to place the symbol.

In an observational study [1], we compared how university students with no experience in inputting mathematics in a computer entered expressions with this model, with a structure-based editor, and by handwriting. It was shown that, unlike a structure-based editor, users had no difficulty quickly grasping the diagram-editor user-interface model, and they wrote expressions in the same order with it as they did by hand.

In terms of manipulation of mathematical expressions, because the expressions are diagrammatic, they can easily be modified, copied, and pasted, just as is the case with diagrams in a vector-based editor. For example (see Fig. 5), if a user clicks on a re-sizeable symbol, draggable resize widgets are added to

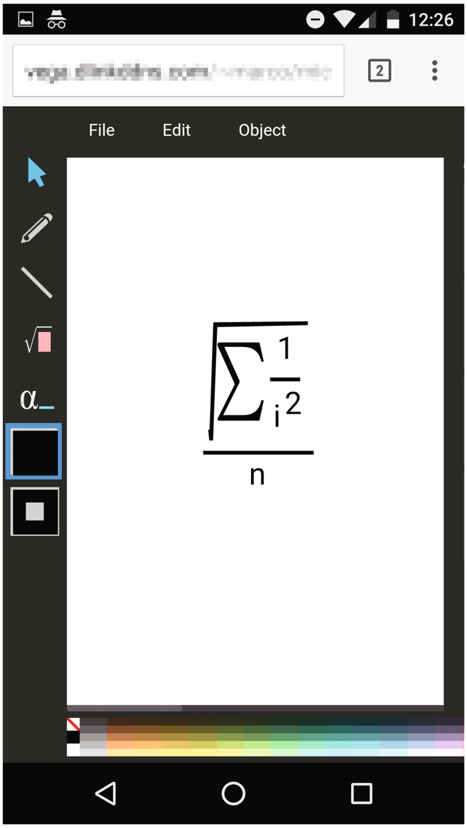


Fig. 4. Screenshot of the smartphone version of the iCE: interface for Collaborative Equations

the element. But if a selection is dragged out, all elements within that selection are temporary grouped together. That grouping could then easily be deleted or dragged to another location.

While the structure-based and pen-based input of mathematics involves a dichotomy between entry and manipulation of expressions, entry and manipulation with the diagram editor uses the same model. It has been shown that users are quickly able to grasp this and manipulate expression with relative ease. To invert a fraction, for example, users could select the numerator by dragging a selection box and then moving it to a new location temporarily. At this point, the denominator is moved to the numerator and the old numerator is moved into the denominator position. Unlike with a structure-based editor, each symbol is at the same layer level, so even deeply nested structures can be modified by clicking on or dragging components.

One weakness of this model is that users have been observed [11] to spend about 25% of their time making the diagram look ‘prettier’, which, of course,

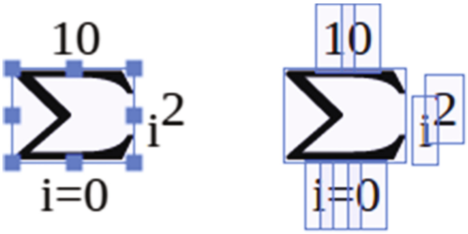


Fig. 5. Screenshot of a diagram-editor based mathematical input model showing a single selected symbol (right) and selection of multiple symbols (right).

benefits the structural analysis algorithm in no way. So, the model could perhaps be improved by incorporating more structural analysis as the user is creating or modifying an expression. When symbols are in a correct position, for example, they could become ‘stickier’, just as with the snap-to-baseline that is already used. So, while users would always have the freedom to place symbols anywhere they want, symbols would tend to be attracted to their ‘correct’ location. This would likely speed up the input process.

4 Conclusion

While the computer input of mathematical expressions is a well-studied topic, few studies have focused on the manipulation of mathematical expressions. This is an important topic, given the pedagogical potential of building online tools for mathematical communication and collaboration, particularly for students who are novice users. In this paper, we have discussed existing input models and showed that they are limited in their ability to allow for easy editing and manipulation of mathematical expressions. And we have argued that a diagram-editor model for mathematical expression entry allows not only for easier input, but also for an intuitive approach to manipulation. This is an avenue of research that would benefit from further study.

Acknowledgments. The work of the authors was partially supported by a research grant from eCampus Ontario.

References

1. Gozli, D.G., Pollanen, M., Reynolds, M.: The characteristics of writing environments for mathematics: behavioral consequences and implications for software design and usability. In: Carette, J., Dixon, L., Coen, C.S., Watt, S.M. (eds.) CICM 2009. LNCS (LNAI), vol. 5625, pp. 310–324. Springer, Heidelberg (2009). doi:10.1007/978-3-642-02614-0_26
2. Helvie-Mason, L.: Office hours: ‘there’s an app. for that?!’: student perceptions of faculty channels for out-of-class communication. Int. J. Instr. Technol. Distance Learn. **9**, 31–38 (2012)

3. Hooper, J., Pollanen, M., Teismann, H.: Effective online office hours in the mathematical sciences. J. Online Learn. Teach. **2**(3), 187–194 (2006)
4. Krupnick, C.G.: Women and men in the classroom: inequality and its remedies. On Teach. Learn. **1**(1), 18–25 (1985)
5. Labahn, G., Lank, E., MacLean, S., Marzouk, M., Tausky, D.: Mathbrush: a system for doing math on pen-based devices. In: The Eighth IAPR International Workshop on Document Analysis Systems, DAS 2008, pp. 599–606. IEEE, September 2008
6. Mazur, E.: Farewell, lecture? Science **323**(5910), 50–51 (2009)
7. Nadler, M.K., Nadler, L.B.: Out-of-class communications between faculty and students: a faculty perspective. Commun. Stud. **51**(2), 176–188 (2000)
8. Padovani, L., Solmi, R.: An investigation on the dynamics of direct-manipulation editors for mathematics. In: Asperti, A., Bancerek, G., Trybulec, A. (eds.) MKM 2004. LNCS, vol. 3119, pp. 302–316. Springer, Heidelberg (2004). doi:10.1007/978-3-540-27818-4_22
9. Perry, A.P.: Decreasing math anxiety in college students. Coll. Student J. **38**, 321–325 (2004)
10. Pollanen, M., Wisniewski, T., Yu, X.: XPRESS: a novice interface for the real-time communication of mathematical expressions. In: Proceedings of MathUI , 8 pages (2007)
11. Pollanen, M., Reynolds, M.: A model for effective real-time entry of mathematical expressions. Res. Reflections Innovations Integr. ICT Educ. **3**, 1235–1512 (2009). Formatex
12. Pollanen, M., Hooper, J., Cater, B., Kang, S.: A tablet-compatible web-interface for mathematical collaboration. In: Hong, H., Yap, C. (eds.) ICMS 2014. LNCS, vol. 8592, pp. 614–620. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44199-2_92
13. Pollanen, M., Hooper, J., Cater, B., Kang, S.: Towards a universal interface for real-time mathematical communication. In: Proceedings of MathUI 2014: CICM-WS-WiP 2014, vol. 1186, 12 pages (2014)
14. Smithies, S., Novins, K., Arvo, J.: A handwriting-based equation editor. In: Graphics Interface, vol. 99, pp. 84-91, June 1999
15. Zanibbi, R., Blostein, D., Cordy, J.R.: Recognizing mathematical expressions using tree transformation. IEEE Trans. Pattern Anal. Mach. Intell. **24**(11), 1455–1467 (2002)