In ‘Modelling and Mathematics Education. ICTMA 9: Applications in Science and Technology. J. f. Matos et al (Eds.). Horwood Publishing 2001. pp. 161-170.

What should be asked of a computer program for mathematical modelling in primary/lower secondary school?

Inge B. Larsen (ibl@dpu.dk)
The Danish University of Education, Dept. of Curriculum Research, Mathematics Unit, Denmark

A very important question, with the present focus on mathematical modelling in the primary/lower secondary school, is: What kind of e-tools will promote children’s ability to build models?

Partly because it is widely and readily available the spreadsheet has become a very popular tool for building algebraic models. However, it can be debated whether the spreadsheet programs in their current format are properly suited for mathematics education at the primary/lower secondary level.

A glaring weakness in today’s spreadsheet programs, from a pedagogical point of view, is the fact that the mathematics describing the model is hidden. In other words the formulas are not directly visible in a spreadsheet. In a spreadsheet model you can manipulate the input and observe the effect on the output, but you can’t directly see the mathematical relations between them.

Ideas for an e-tool for modelling in school, that builds on the core idea in a spreadsheet program, but in which the model is directly visible, is presented. Activities for a first implementation of the ideas (a program called VisiComp) has been developed and used successfully in school on the third to eight grade level.

THE TRADE-OFF BETWEEN POWER AND EASE OF HANDLING

There are two important questions to be asked of any tool:

– What power does it provide?
– How easy is it to handle?

In general there is a trade-off between the two: power and ease of handling. The more power you want, the more difficult it will be to handle the program.

When it comes to e-tools for mathematics education, this trade-off between power and ease of handling is crucial. Handling a program, i.e. learning the programs' user interface has nothing to do with mathematics. So if much of the time and effort goes into learning how to handle the e-tools, less time will be devoted to learning mathematics. Furthermore, there is the danger that learning how to handle the program will be confused with learning mathematics. Another thing that ought to be kept in mind is that the user interface of computer systems and programs change rapidly, so learning to handle a specific program may be of a very limited value.

It could from a random survey of articles (examples: Kieran, 1998 and Sutherland, 1993) in international journals be postulated that the sort of program mostly used for algebraic modelling in primary and lower secondary school is the spreadsheet program. Two students at Harvard Business School created the first spreadsheet program VisiCalc more than 20 years ago (Carland, 1988) and the main idea of the first spreadsheet program has remained unchanged in the many different spreadsheet programs that followed it. VisiCalc was created for the business world, and that is still the main target for spreadsheet programs. But since spreadsheet programs were there, and they were easier to handle than a programming language, they also found their way into school mathematics.

Rosamund Sutherland (1993) concludes: 'This study shows that working with a spreadsheet can help pupils develop algebraic ideas which they can also use in a paper setting. The transfer from spreadsheet to traditional algebraic symbolism is not as difficult as many people have predicted.'

This is a very promising result, but maybe we can do even better, if we try from the viewpoint of mathematics education to identify the strengths and the weaknesses of spreadsheet programs? It is certainly worthwhile to address this question. Every time we make the handling of a program easier and more purposeful, we gain in time that can be devoted to learning mathematics, and no doubt we also get rid of a lot of frustrations both for the teacher and the students. Right now, it seems as if we are adapting to the spreadsheet programs (made for the business world) instead of the programs adapting to our needs in mathematics education. This is indeed strange (and short-sighted) considering the resources that might be saved by changing the programs instead of all the teachers and students.

AN ATTEMPT TO IDENTIFY SOME UNFORTUNATE FEATURES IN SPREADSHEET PROGRAMS AS SEEN FROM AN EDUCATIONAL POINT OF VIEW

1. Lack of visibility

The first and most serious objection to spreadsheet programs is their lack of visibility. The first spreadsheet program was called VisiCalc for Visible Calculations, and compared to what happens when using a programming language, things are definitely more visible (or better: more accessible) in a spreadsheet program.

However, the algebraic model in a spreadsheet program is expressed through the formulas, and the formu las unfortunately are hidden and only their values are shown. Pointing at the cell it is placed in can show a single formula in a spreadsheet model, but it is hard to comprehend the whole model this way. The spreadsheet can be set to show all its formulas, but then it is no longer possible to see their values, and unless the cells through tedious work has been given informative names, the model can be very difficult to decode. In Excel you can ask for arrows that point out dependencies, but they do not tell you the kind of the dependencies.

The most important parts of the spreadsheet, from the viewpoint of mathematics education, are the formulas that go into building the algebraic model, and it is therefore a serious deficiency that they are not visible. This deficiency can be used for activities like ‘Find the formula’ (Healy, 1991), but that doesn’t make up for all the other situations where the lack of visibility is a serious drawback. It is like blindfolding the students when they work with the algebraic model. They put in a formula to express a certain relationship, and immediately the formula is hidden and only a number (its value) is visible.

When you look at a spreadsheet, you see text and numbers. Presumably, some of the numbers are input data and others are output data produced by an algebraic model. But to separate the input data from the output data is not possible just by looking. Needless to say, this lack of transparency makes the building and using of models hard, especially in a learning situation.

The Danish school system has a final written test in mathematics at the end of lower secondary level. A student can choose to use a spreadsheet program for this test, but the hand-in has to be on paper. This creates an unhappy situation, since the student is supposed to describe in detail how the results were obtained. One way to overcome this obstacle has been to make a copy of every cell with a formula and then place the copy in an adjacent cell as text. The lack of visibility in a spreadsheet creates this ridiculous procedure.

2. Variables and their names

A variable in standard notation, like x in $f(x)=2x+3$, can take on many different values, but at the moment, it doesn’t seem to have a special value. This makes it somewhat abstract and often incomprehensive for the student.

A variable in a spreadsheet, like A2 (the name of the cell in the second row and first column), is always supposed to have a value. In fact, if you don’t give it a value, it will be assumed to have the value 0 (at least in Excel and Works spreadsheets). The fact, that you can easily change the value of a cell, makes us think of it as a variable. One might expect that a variable, that always has an actual value, is less abstract than the traditional one.

Variables in traditional mathematics have names like x, y and n. Names that make formulas and expressions very short and therefore easy to form a general view of and to write and rewrite. But also names that don't convey much information about the roles of the variables used to express a certain relationship.

Variables in spreadsheets have names like A7, K36 and AB12. Such a name has two functions: it is a variable name that can be used in formulas and it is a reference to a certain position in the spreadsheet, where its value can be found. Names in a formula may be equipped with $-signs like in the formula: =C$5*$B6+C3 to prepare the formula for a certain type of copying. It is useful to be able to copy formulas in different ways, but here it certainly does not add to the readability of the model. It is possible in Excel spreadsheets to give the cells more informative names, but it demands an extra effort from the user, and it doesn't seem to be a straightforward thing to do.

Students learning about building algebraic models should at least be able to choose informative names for variables and to view the build in dependencies and their effects at all time.

3. *Storing of guesses used in trial-and-error situations*

An algebraic model in a spreadsheet form gives you an easy and fast way to explore the model's response to different input values and this feature can be used to solve problems that would otherwise be out of reach. However, the user has to keep track of the tried input values and the output they produced. It would be helpful both for the teacher and the student if the program itself could keep a record of the guesses and their effect, and if a graph of the record could be drawn. In this way it would be possible to illustrate the interaction between the algebraic model, the data collected from the algebraic model and the graphic representation of these data.

4. *Charts and notation*

The spreadsheet programs have an abundance of chart types, and each of them can be formatted in many different ways. Learning all these facilities is time-consuming. Many of the types of charts in for example Excel have nothing to do with school mathematics. Learning to ignore them takes time too.

The notation used in connection with spreadsheet deviates from the standard notation in school mathematics. How will spreadsheet terms like data series, category names etc. fit into the standard notation used in school mathematics? And what about the very peculiar use of the equal sign in a spreadsheet formula? Are we in school algebra going to abandon the traditional notation and use spreadsheet notation instead? Or will the students have to learn both notations and their relationships?

5. *Copying with relative/absolute references*

Copying with relative/absolute references could be considered a replacement for the loop constructions that are found in programming languages. It is very useful but also a rather difficult facility to grasp, and maybe in most cases the recording facility mentioned in point 3. above would be a more straightforward substitution for copying.

It seems as if we are trying to adjust teachers, students and school algebra to an electronic tool, that was developed with the business world in mind, instead of looking for an electronic tool, that would fulfil the demands of the learning of school algebra. If we move to algebraic modelling at the secondary and higher level, the situation is very different. There we have the Computer Algebra Systems (CAS) that were developed by mathematicians and for the use in mathematics.

TOWARDS AN E-TOOL FOR ALGEBRAIC THINKING

The INFA-project at the Danish University of Education has for many years been concerned with the development of e-tools for mathematics at the primary and secondary level. The first edition of an e-tool for algebraic thinking, miniREGN, dates back to 1994. The second edition came out in 1995. It was produced the following year in an English version called miniCALC. The third edition, after several delays, was finished in December 2000 and is called VisiRegn. In English it will be called VisiComp for Visible Computations – visible in a way that the original spreadsheet program VisiCalc and its followers never were.

VisiComp can be described as a spreadsheet with just one column. However, this column has been divided into four sub columns. In the first column a (variable) name can be written. In the second column a formula/an algebraic expression possibly involving names used in the lines above can be written. The value of the formula will automatically be shown in the third column and in the fourth column a unit can be placed. This allows for making a model visible at all times. It is possible to copy with relative and absolute references as in a spreadsheet program, and it is also possible to make a table in which data from the model is collected, and to make different charts.

This is by no means the final solution to finding an appropriate program for working with calculations and algebraic model building at the primary and lower secondary level, but we do consider it an important step in the right direction.

EXAMPLES ON USING VisiComp:

The few examples that follow have to be described very briefly, but hopefully they will convey an idea of how the program can be useful in school algebra. The aim is to give the students at primary and lower secondary level the power to:

- easily make, use and compare algebraic models
- easily have data from an algebraic model illustrated in tables and graphs

1. From arithmetic to algebraic modelling

The price of an item is 300 Danish kroner. To this you have to add the sales tax, which in Denmark is 25%. How much will you have to pay?

	Name	Expression	Value	Unit
A1	price	300	300.00	kr.
A2	tax	300*25/100	75.00	kr.
A3	pay	300+75	375.00	kr.

The solution above is made in VisiComp. Each line is split up in 3 parts corresponding to what students (at least in Denmark) for many years have been told to do: “First you write what you are going to find, then you write how you’ll find it, and finally you write the result and don’t forget the unit.” In VisiComp the student only has to do the first, the second and the fourth step, the third step follows automatically from the expression in the second step. VisiComp is in this example used as an advanced pocket calculator.

The price of another item is 560 kroner. With the sales tax added, what will you have to pay for it? To answer this with the set up above it is necessary to change 4 numbers in the 3 expressions. That is tedious, but luckily a name in VisiComp is valuable in more than one sense. It doesn’t just tell what is going to be found, it actually has a value, namely the value given in the same line. That means, that the set up above can be turned into a model by using the names instead of the values:

Name	Expression	Value	Unit
price	560	560.00	kr.
tax	price*25/100	140.00	kr.
pay	price+tax	700.00	kr.

The advantage of turning a ‘number set up’ into a model is here obvious and easy done, and our experiences with schoolchildren age 10-14 years using the program are promising. It looks as if this could be an electronic bridge helping the student to go from arithmetic to algebraic modelling.

The power of the e-model becomes even more obvious when faced with questions like: If you can at most pay 447,50 kr., what is the highest price of an item you can buy?

A question like this traditionally demands that you can calculate backwards or set up an equation and solve it; both are methods that many students find difficult. But with the model at hand you can use a straightforward trial-and-error approach. Data from the model are collected in a table, as shown below. A T to the left of price and pay indicate that these names shall be included in the table to the right. Every time a price is inserted in the first line it will be collected in the table together with its corresponding pay value. The table is extremely useful in guiding the students trial-and-error attempts.

Model

T	*		Name	Expression	Value	Unit
T		A1	price	358	358.00	kr.
		A2	tax	price*25/100	89.50	kr.
T		A3	pay	price+tax	447.50	kr.
		A4				
		A5				
		A6				

Table

price	pay
300.00	375.00
350.00	437.50
360.00	450.00
355.00	443.75
357.00	446.25
358.00	447.50

2. Algebraic modelling of a ‘real life’ situation

The following is just one example of the many simple algebraic models we encounter in our daily life. It is from an article (about keeping in shape by walking) in a popular Danish health journal:

“… Your pulse while walking should be 70 percent of your maximum pulse. Your maximum pulse is 220 beats per minute minus your age. If you are 50 years old, then your maximum pulse is 220 minus 50 = 170 beats per minute. You find 70 percent of 170 by dividing with 100 and multiplying with 70, that gives 119. If you are 50 years old, then you should have a heartbeat of 119 per minute during the walking.”

	Name	Expression	Value	Unit
A1		"Recommended pulse during walking for		
A2		"keeping in shape.		
A3	age	50	50	years
A4	maxpuls	220-age	170	beat/m
A5	walkpuls	maxpuls*70/100	119	beat/m
A6				
A7		"The model given in 1 line instead of 2:		
A8	wpulse1	(220-age)*70/100	119	beat/m
A9		"Simplification of this model:		
A10	wpulse2	-0.7*age+154	119	beat/m

First the student has given the model step by step in 2 lines (plus the input line). Then the student was asked to give the model in just one line. Finally, using knowledge of how to simplify such an expression the student has typed in the model in the traditional form of a linear function.

Even though the computers don’t care about the length of an expression, it is still important that the students know how to simplify expressions. In the example above the student needs this skill:

- to be able to figure out that two models are identical, even if they don’t look the same
- to be able to identify a model as belonging to a family of models with known properties

3. Simplifying expressions

That algebra can be useful is often illustrated by using algebra to reveal ‘the magic’ in games like:

- Think of a number (and write it down)
- Add 2
- Multiply with 2
- Subtract 4
- Divide by 2 (and write down the result)

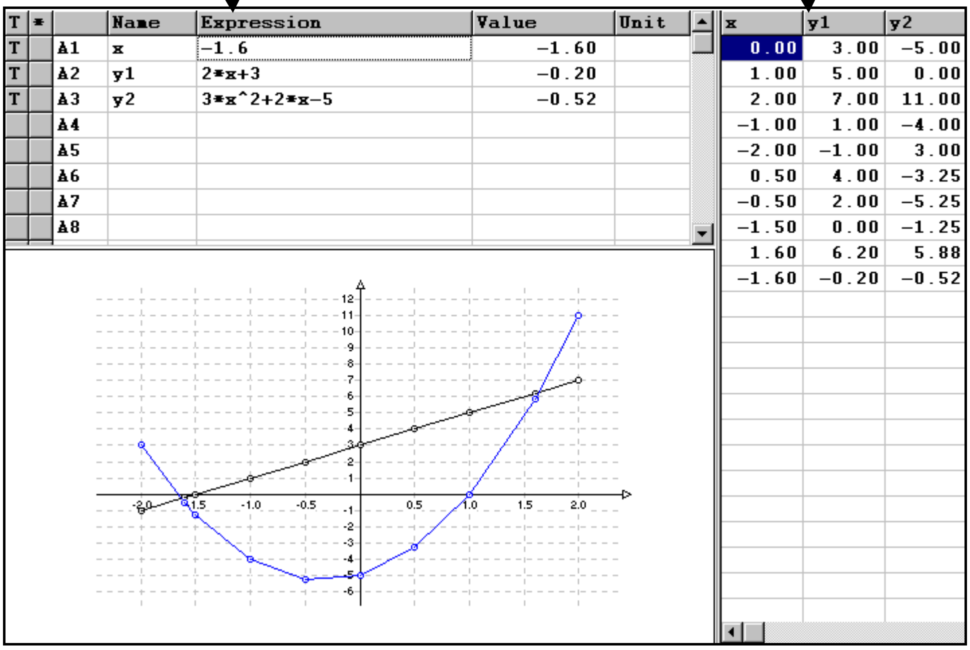
Below the model is first written in a step-by-step fashion. With this model it’s easy to explore, if the model always returns the starting number, even with starting numbers like 0, -23 and 2,738. Next the student has written the model in just one line and has checked, if this new version for different input values returns the same result as the first model. If this is not the case, the student will know that something is wrong. If the two versions of the model give the same output for many different values, it might indicate, but of course not prove, that the two models are identical. In other words the program can provide a necessary, but not a sufficient condition for the models being identical. In this way the student can continue to check, if she is on the right track doing the next simplifications.

	Name	Expression	Value	Unit
A1	x	7	7	
A2		x+2	9	
A3		A2*2	18	
A4		A3-4	14	
A5		A4/2	7	
A6		"The above model in just one line:		
A7		((x+2)*2-4)/2	7	
A8		"Simplifying this gives:		
A9		(x*2+4-4)/2	7	
A10		x*2/2	7	
A11		x	7	

4. A close correspondence between the algebraic model, data from the model and graphic picture of the data

As seen on the following page, the program gives an immediate visual presentation of the connection between the algebraic model, the table with data that the student collect from the model and the graphical picture of the data from the table. The different input values that the student has provided and their corresponding output values are recorded in a table to the right, and the record is then illustrated graphically below. Thus there is a close correspondence between the model, the table and the graph. Numbers fed into the model appears immediately in the table together with the corresponding output data, and at the same time the correspondent points appear in the graph. At first only the points are shown, but it is possible to ask for having the points connected with lines. With the immediate feedback from the graph the student gets ideas for the next values to be fed into the model. It is also possible to change the graph by deleting rows in the table.

The algebraic model *Record of trial-and-error use of the model*



Graphical illustration of the record

CONCLUSION
The computer will play an increasing role in mathematics education, not just in connection with the setting of goals and curriculum but also in connection with methods of learning. A computer program sets the agenda in a more decisive way than a book. Therefore it is important that we constantly evaluate, how we can improve the available programs.

Hopefully more attention and more resources will be devoted to the very important subject of creating appropriate e-tools for mathematics education.

REFERENCES

Ainley, Janet (1996) '*Purposeful Context for Formal Notation in a Spreadsheet Environment*'. Journal of Mathematical Behavior 15, 405-422.

Carland, Jo Ann C., Carland, James W. and Aby, Jr., Carroll D. (1988) '*Spreadsheets: Placebo or Panaceas*'. Journal of Research on Computing in Education, Fall, 112-119.

Healey, Lulu and Sutherland, Rosamund (1991) '*Exploring Mathematics with Spreadsheets*'. Simon and Schuster.

Kieran, Carolyn (1992). 'The Learning and Teaching of School Algebra' in Douglas A. Grouws (Ed.) *Handbook of Research on Mathematics Teaching and Learning.* Macmillan, 390-419.

Kieran, Carolyn (1998). 'The changing face of school algebra' in C. Alsina, J. Alvarez, B. Hodgson, C. Laborde and A. Pérez (Eds.) *8th International Congress on Mathematical Education, Sevilla 1996, Selected Lectures.* S.A.E.M 'THALES', 271-286.

Larsen, Inge B. (1996a) '*Introduction to miniCALC*'. The Royal Danish School of Educational Studies, INFA, MI 111.

Larsen, Inge B. (1996b) '*Regneprogrammer - en analyse af faglige og paedagogiske muligheder*'. The Royal Danish School of Educational Studies, INFA, MI 118.

Larsen, Inge B. and Sadolin, Viggo (1998) '*Matematik I Skolen - It & Laering. MISIL-projektet.. Rapport 97/98*'. The Royal Danish School of Educational Studies, INFA, MI 139.

Sutherland, Rosamund (1993) '*Symbolising through Spreadsheets*'. Micromath, Spring, 20-22.