

TrueGrid: Code in a Grid and Grids in the IDE

Felienne Hermans and Tijs van der Storm

TODO
f.f.j.hermans@tudelft.nl
todo

Abstract. Spreadsheet systems are live programming environments. Both the data and the code are right in front you, and if you edit either of them, the effects are immediately visible. Unfortunately, spreadsheets lack mechanisms for abstraction, such as classes, function definitions etc. Programming languages excel at abstraction, but most mainstream languages or integrated development environments (IDEs) do not support the interactive, live feedback loop of spreadsheets. As a result, exploring and testing of code is cumbersome and indirect.

In this paper we propose a method to bring both worlds closer together. We propose TrueGrid: an approach where spreadsheet cells can be programmed not with formulas but a full featured programming language (JavaScript).

We then describe TrueGrid within two different worlds: the spreadsheet world, where it adds the benefits of source code, including added abstractions, syntax highlighting, version control. We then describe TrueGrid within the IDE, where it brings the benefits of the liveness of spreadsheets which will support rapid application development.

1 Introduction

1.1 Spreadsheets

Spreadsheets are very popular tools for end-user programming **cite proof**. Their formula language is easy to learn and their grid interface is inviting. Apart from this, spreadsheets have the interesting property of being *live* programming. First proposed as a design principle by Maloney and Smith [2], liveness indicates that a user interface is always active and reactive. According to [2] in a live interface “objects respond to user actions, animations run, layout happens, and information displays are updated continuously”. More recently live programming has found its way to the public eye, among others by Bret Victor in his talk ‘Inventing on Principle’ [4]. Figure 1, taken from Victor’s talk, illustrates the idea of live programming: on the right, we have source code and on the left, we have the result of that code, in this case: a tree. Modifying the code will immediately affect the tree.

This liveness is one of the core features of most spreadsheet systems. When a users enters a formula and presses enter, they see the result, without any effort

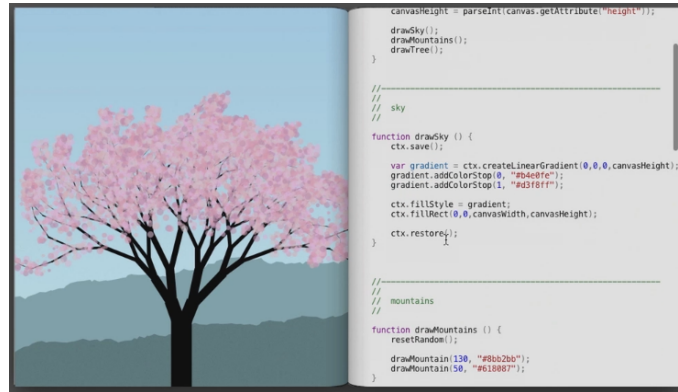


Fig. 1. Live programming: on the right the source code and on the left its instantiation of the code which changes immediately when the code is updated, screenshot from [4].

such as compilation. Liveness of spreadsheets powers a spreadsheet’s flexibility, often praised as their key success factor.

However, spreadsheets have a number of downsides too, for example the lack of possibilities to abstraction. Instead of creating objects and methods on them, spreadsheet users mimic abstraction by copy-pasting formulas **references naar LIVE paper**. Secondly, the editor that Excel and other spreadsheet systems provide is very limited, for example lacking syntax highlighting and not inviting proper structuring of formulas.

1.2 Code

On the other hand, there is source code. Most existing programming languages do support abstraction, and modern editors help developers understand and structure their code. However, source code does not have liveness **apart from a few recent initiative that have not been adopted mainstream, Tijs, hedd gij dr paraat?**

2 TrueGrid

In this paper, we will describe an approach bridging the gap between programming and spreadsheets and describe its implications in both worlds. The basic concept of TrueGrid is a grid which can be programmed in a fully featured programming language, where the grid and the code are juxtapositioned, like in a spreadsheet, as demonstrated in Figure 2. Furthermore, like a spreadsheet, TrueGrid is live, i.e. on a change of data or code, the grid is updated.

In our example, we use JavaScript as a language, however, the idea itself is not limited to one programming language, in fact, one can even imagine a grid in which different programming languages live.

input1	input2	test
1	3	?
2	5	?
2	7	?

```
function test(p1, p2) {return p1 * p2;}
```

Fig. 2. A True Grid **make ss** with syntax highlighting, and add explanation

3 TrueGrid for Spreadsheet Users

As an implementation of TrueGrid for spreadsheet users, we hypothesize that it will be embedded within an existing spreadsheet system, allowing users to use advanced features like PivotTables and charts on top of their TrueGrid. For spreadsheet users, using TrueGrid over spreadsheets presents several benefits. For example, a programming language enables the use of modern editors. In the above example we have used Ace **add link** providing syntax highlighting and syntax errors for free. Furthermore, the textual form of the code allows for easy diffing, merging enabling more mature version control on grids. While learning a new programming language can be challenging, there are spreadsheet developers working with VBA now, which is a fully featured language, and we envision TrueGrid having a lower threshold as the liveness and juxtaposition will ease understanding.

4 TrueGrid for Developers

Like spreadsheet users, developers often work with (example) data when programming. In modern IDEs, it is not possible to load data and then program alongside it. This makes it difficult for quickly prototype simple data analysis projects. A TrueGrid embedded in the IDE

Using TrueGrid, a programmer you can inspect data and play with an initial implementation, before having to write interfaces or even unit tests. We envision this is a useful tool very early in the development process, when you have some data and only a vague idea of where the project is going.

For example, consider the situation where a developer has a list of number that you want to manipulate **We need a better example here** Before she even wants to write tests, the developer might want to explore the dataset a bit, for example, by sorting, calculating the mean or outliers. These type of activities

currently have to be done outside of the IDE, for example in Excel or using a scripting language like R. Using TrueGrid this activity can be performed within the IDE.

5 Related Work

tsja

References

1. Burnett, M.M.: Visual Programming. In: Wiley Encyclopedia of Electrical and Electronics Engineering. John Wiley & Sons, Inc. (2001), <http://onlinelibrary.wiley.com/doi/10.1002/047134608X.W1707/abstract>
2. Maloney, J.H., Smith, R.B.: Directness and Liveness in the Morphic User Interface Construction Environment. In: Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology. pp. 21–28. UIST '95, ACM, New York, NY, USA (1995), <http://doi.acm.org/10.1145/215585.215636>
3. Shneiderman, B.: Direct Manipulation: A Step Beyond Programming Languages. *Computer* 16(8), 57–69 (Aug 1983)
4. Victor, B.: Inventing on principle (2012), <http://vimeo.com/36579366>