

Haxcel: A Spreadsheet Interface to Haskell written in Java

Johan Malmström
Dept. of Computer Science and Engineering
Mälardalen University

`jmm98002@idt.mdh.se` `http://www.mds.mdh.se/~dal98jmm`
`http://www.mrtc.mdh.se/projects/Haxcel/`

2002-12-17

Abstract

The spreadsheet paradigm offers a fast interactive loop, where the effects of updates to definitions and data are immediately visible. This makes the paradigm attractive for program development, where redefinitions can be immediately tested and the results displayed. We have designed a simple, compiler-independent spreadsheet interface to Haskell, where cells host Haskell definitions. Spreadsheets are also used for high-level array calculations. In order to meet that demand we present an extended array library for Haskell, which provides a number of typical array-language facilities. Together, the interface and the array library provide an interactive environment that can be used both for development of general Haskell code and for array-oriented spreadsheet calculations.

Introduction

Spreadsheets History and a introduction to common spreadsheets.

Introduction

Spreadsheets History and a introduction to common spreadsheets.

Haskell A brief introduction to the functional programming language Haskell

Introduction

Spreadsheets History and a introduction to common spreadsheets.

Haskell A brief introduction to the functional programming language Haskell

Haxcel Our prototype.

Introduction

Spreadsheets History and a introduction to common spreadsheets.

Haskell A brief introduction to the functional programming language Haskell

Haxcel Our prototype.

Xarray A extended array library to Haskell which provides a number of typical array-language facilities.

Introduction

Spreadsheets History and a introduction to common spreadsheets.

Haskell A brief introduction to the functional programming language Haskell

Haxcel Our prototype.

Xarray A extended array library to Haskell which provides a number of typical array-language facilities.

Conclusions and future work

Spreadsheets

VisiCalc introduces spreadsheets for computers 1979 for Apple II

Spreadsheets

VisiCalc introduces spreadsheets for computers 1979 for Apple II

Common spreadsheet applications like Microsoft's *Excel* and P & L Software's *Mesa* share the same setup:

- 2-D area
- Cells, Rows, Columns
- Value, Formula, Format

Spreadsheets

VisiCalc introduces spreadsheets for computers 1979 for Apple II

Common spreadsheet applications like Microsoft's *Excel* and P & L Software's *Mesa* share the same setup:

- 2-D area
- Cells, Rows, Columns
- Value, Formula, Format

Applications without traditional programming skills

Spreadsheets

VisiCalc introduces spreadsheets for computers 1979 for Apple II

Common spreadsheet applications like Microsoft's *Excel* and P & L Software's *Mesa* share the same setup:

- 2-D area
- Cells, Rows, Columns
- Value, Formula, Format

Applications without traditional programming skills

Definition-Eval-Display loop gives immediate feedback

The screenshot shows the Haxcel spreadsheet application window titled 'mesa.mesa'. The interface includes a menu bar with 'Save', 'Inspectors', 'Formulas', 'Reports', 'Charts', and 'Print'. A 'Sheet 1' tab is active. The formula bar displays the formula $=(D4 - D2) / D4 * 100$. The spreadsheet grid shows the following data:

	A	B	C	D	E	F	G	H	I
1	Serie	Start	End	Growth	Decrease				
2	1	0.0070E+	0.25	0.89					
3	2	0.0070E+	0.22	0.86	-3.98				
4	3	0.0070E+	0.17	0.79	-13.15				
5	4	0.0070E+	0.0540E+	0.51	-75.01				
6	5	0.0070E+	0.0120E+	0.13	-563.3718E+				
7	6	0.0070E+	0.0060E+	-0.0385E+	2419.5167E+				
8	7	0.0070E+	0.0050E+	-0.0841E+	1162.6585E+				
9									
10									
11									
12									
13									
14									
15									

Problems in common spreadsheet applications

- Insertion of rows and columns

Problems in common spreadsheet applications

- Insertion of rows and columns
- Hard to do simple data manipulations

Problems in common spreadsheet applications

- Insertion of rows and columns
- Hard to do simple data manipulations
- Code reuse and function definitions

Problems in common spreadsheet applications

- Insertion of rows and columns
- Hard to do simple data manipulations
- Code reuse and function definitions
- No local variables

Problems in common spreadsheet applications

- Insertion of rows and columns
- Hard to do simple data manipulations
- Code reuse and function definitions
- No local variables
- Limited number of data types

Problems in common spreadsheet applications

- Insertion of rows and columns
- Hard to do simple data manipulations
- Code reuse and function definitions
- No local variables
- Limited number of data types
- No recursion in languages

Haskell

- Functional programming language

Haskell

- Functional programming language
- Haskell 98, Haskell Report

Haskell

- Functional programming language
- Haskell 98, Haskell Report
- Modules and declarations

Haskell

- Functional programming language
- Haskell 98, Haskell Report
- Modules and declarations
- hmake and hi works with nhc, ghc and hbc.

Haskell

- Functional programming language
- Haskell 98, Haskell Report
- Modules and declarations
- hmake and hi works with nhc, ghc and hbc.
- Hugs and runhugs fast interpreter.

Haskell 2

A very simple example of a Haskell module with one declaration.

```
module Max2 where

-- max2 returns the largest of two integers
max2 :: Int -> Int -> Int
max2 x y
= if x >= y then x else y
```

Haxcel

- Intro
 - ★ Java application
 - ★ Independent from Haskell compilers
 - ★ Multi-window spreadsheet

Haxcel

- Intro
 - ★ Java application
 - ★ Independent from Haskell compilers
 - ★ Multi-window spreadsheet

- Single Module

Haxcel

- Intro
 - ★ Java application
 - ★ Independent from Haskell compilers
 - ★ Multi-window spreadsheet
- Single Module
- Create declarations

Haxcel

- Intro
 - ★ Java application
 - ★ Independent from Haskell compilers
 - ★ Multi-window spreadsheet
- Single Module
- Create declarations
- Export Module to conventional Haskell file

Simple example

Xarray

- Array language

Xarray

- Array language
- Some of the basic features of Xarray:
 - ★ Computing bounds of arrays with functions `join` and `meet`
 - ★ Scalable arrays is provided through the type of the extended array.
 - ★ Projection operations
 - ★ Dimensions operations, makes matrices out of vectors and splits matrices into vectors

Budget example

Conclusions

- Compiler-independent spreadsheet interface for Haskell

Conclusions

- Compiler-independent spreadsheet interface for Haskell
- Over comes some of the problems in conventional spreadsheets

Conclusions

- Compiler-independent spreadsheet interface for Haskell
- Over comes some of the problems in conventional spreadsheets
- Haskell is good for spreadsheet programming

Conclusions

- Compiler-independent spreadsheet interface for Haskell
- Over comes some of the problems in conventional spreadsheets
- Haskell is good for spreadsheet programming
- Fast with runhugs, annoyingly slow with hmake/nhc

Conclusions

- Compiler-independent spreadsheet interface for Haskell
- Over comes some of the problems in conventional spreadsheets
- Haskell is good for spreadsheet programming
- Fast with runhugs, annoyingly slow with hmake/nhc
- Import of .hs file, literate scripts

MRTC

<http://www.mrtc.mdh.se/projects/Haxcel/>