Detecting and Visualizing Inter-worksheet Smells in Spreadsheets

Felienne Hermans (@felienne)
Code smells
How did we get this idea?
By studying what business does in practice
My image of reality
Reality
Excel is running companies!
95% of all U.S. firms use spreadsheets for financial reporting
90% of all analysts in industry perform calculations in spreadsheets.
50% of spreadsheets form the basis for decisions
Spreadsheets are filled with precious, but unobtainable information.
Automatically Extracting Class Diagrams from Spreadsheets

Felienne Hermans, Martin Pinzger, and Arie van Deursen
Delft University of Technology
{f.f.j.hermans,m.pinzger,arie.vandeursen}@tudelft.nl

Abstract. The use of spreadsheets to capture information is widespread in industry. Spreadsheets can thus be a wealthy source of domain information. We propose to automatically extract this information and transform it into class diagrams. The resulting class diagram can be used by software engineers to understand, refine, or re-implement the spreadsheet’s functionality. To enable the transformation into class diagrams we create a library of common spreadsheet usage patterns. These patterns are localized in the spreadsheet using a two-dimensional parsing algorithm. The resulting parse tree is transformed and enriched with information from the library. We evaluate our approach on the spreadsheets from the Eases Spreadsheet Corpus by comparing a subset of the generated class diagrams with reference class diagrams created manually.

1 Introduction

To design and implement a software system a high degree of familiarity with the domain of the software is needed. We maintain that a significant portion of this domain knowledge is already contained in spreadsheets, which are widely used in industry. Since spreadsheets’ knowledge is not explicit knowledge in any form, it is the purpose of this paper to make this knowledge explicit.

Spreadsheets were introduced in the early 1980’s with the first spreadsheet tool called VisiCalc. This tool was then followed by SuperCalc and Lotus 123 and later on by Excel which currently is one of the most prominent spreadsheet tools. Since their introduction, spreadsheets are heavily used in industry. A study from the year 2003 shows about 23 million American workers use spreadsheets, which amounts to about 30% of the workforce [24].

Spreadsheets can be a rich source of information concerning the structure of the underlying domain. They contain groups of data, computations over these groups, and data dependencies between them. In our approach, we will attempt to make this structure explicit, by representing it as a class diagram. Groups of data are turned into classes, formula’s into methods, and data dependencies into associations. The resulting class diagram can be used by software engineers to understand, refine, or re-implement the spreadsheet’s functionality.
Extracting it met with some resistance
If Mohammed doesn’t come to the mountain...
Spreadsheet users lack great tool support
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        public static int Mu(int x, ref int y)
        {
            if (x == 1)
            {
                return y;
            }
            else
            {
                // divisible by 2?
                int Rem;
                Math.DivRem(x, 2, out Rem);
                int Stop;
                if (Rem == 0)
We did not start coding!
We performed 27 interviews
Ans asked them: What annoys you?
And what makes you happy?
We found that especially support understanding spreadsheets was missing
Then we started coding
Supporting Professional Spreadsheet Users by Generating Leveled Dataflow Diagrams

Felicijn Hermans
Deit University of Technology
f.f.j.hermans@tudelft.nl

Martin Pinzger
Deit University of Technology
m.pinzger@tudelft.nl

Arie van Deursen
Deit University of Technology
arie.vandeursen@tudelft.nl

ABSTRACT
Thanks to their flexibility and intuitive programming model, spreadsheets are widely used in industry, often for business-critical applications. Similar to software developers, professional spreadsheet users demand support for maintaining and transferring their spreadsheets.

In this paper, we first study the problems and information needs of professional spreadsheet users by means of a survey conducted at a large financial company. Based on these needs, we then present an approach that extracts this information from spreadsheets and presents it in a compact and easy to understand way, with leveled dataflow diagrams. Our approach comes with three different views on the dataflow that allow the user to analyze the dataflow diagrams in a top-down fashion.

To evaluate the usefulness of the proposed approach, we conducted a series of interviews as well as nine case studies in an industrial setting. The results of the evaluation clearly indicate the demand for and usefulness of our approach in case the understanding of spreadsheets.

Categories and Subject Descriptors
H.4.1 [Information Systems]; Office Automation - Spreadsheets; D.2.2 [Software]; Design Tools and Techniques - Flowcharts

General Terms
Design, Experimentation, Human Factors, Languages

1. INTRODUCTION

Spreadsheets are widely used in industry: Winston [23] estimates that 90% of all analysts in industry perform calculations in spreadsheets. Their use is diverse, ranging from inventory administration to educational applications and from scientific modeling to financial systems. The financial business is a domain where spreadsheets are especially prevailing. Panko [18] estimates that 90% of U.S. firms and 80% in Europe, use spreadsheets in some form for financial reporting.

Business analysts using spreadsheets usually have very limited (if any) training as a programmer. In spite of that, they effectively are end-user programmers, and as such face many of the challenges of professional developers, such as identifying faults, debugging, or understanding someone else’s code [15].

This paper aims at providing support for spreadsheet users to tackle these end-user programming challenges. To that end, we first study the problems and information needs of professional spreadsheet users and then present an approach that presents this information in a compact and easy to understand way, with leveled dataflow diagrams.

The context, in which we conduct our research, is formed by the analysts of Robeco, a Dutch asset management company with approximately 1600 employees worldwide, and over €150 billion worth of assets under management. In a survey we conducted among 27 of their analysts, we found that they use Excel for an average of 3 hours a day, underlining the large role spreadsheets play in their daily work. Furthermore, spreadsheets have an average lifetime of more than five years, and individual spreadsheets are used by 13 different analysts on average.

In order to support analysts in their work with spreadsheets, we start by identifying information needs. In particular, we conducted interviews with the same analysts to gather an end user approach, focusing on the views of the end users. In the interviews, solutions transfer scenario (to a new user, to an auditor, and to a professional developer creating custom software) were identified as problematic. In these scenarios, end-users search for a better insight into the dependencies between cells, formulas, groups of cells and worksheets.

To meet these demands, we propose an approach for the automated extraction of dataflow diagrams from spreadsheets. Such diagrams can be used to visualize data, process manipulating data, and dependencies between them. Furthermore, leveled diagrams can be used to accommodate hierarchies, for example for blocks of cells or worksheets.
Users started diagnosing with the dfd’s
This led us to the idea of smells
Inappropriate Intimacy
Feature Envy
Middle man
Shotgun surgery
We formalized this into quantifiable and detectable smells
And validated it with a case study
“Are those worksheets really that connected?”
“Are you sure that arrow is correct?”
“this formula annoys me, I have to go back to the other sheet so many times to look up the references, it makes me dizzy”
Interesting fact about Middle Man: Our definition was ‘wrong’
We had not anticipated Middle Men within worksheets
The most extreme case was a worksheet on which 220 formulas depended, spread over 10 worksheets.
The owner immediately recognized the risk.
Conclusion: principles from software engineering transfer well to spreadsheets
Code smells

We performed 27 interviews
Code smells

We performed an analysis on the codebase and identified several code smells. These smells included long methods, complex conditional statements, and excessive duplication. We found that especially support understanding spreadsheets was missing.

We found that especially support understanding spreadsheets was missing.
and that especially support standing spreadsheets was missing
We formalized this into quantifiable and detectable smells.
We formalized the detectable smell and validated it with a case study.
Detecting and Visualizing Inter-worksheet Smells in Spreadsheets

Felienne Hermans (@felienne)