
**Diogo João Costa Canteiro**

Licenciado em Engenharia Informática

# A Structured Approach to Document Spreadsheets

Dissertação para obtenção do Grau de Mestre em

**Engenharia Informática**

Orientador: Jácome Cunha, Professor Auxiliar,
Universidade Nova de Lisboa

Júri

| | |
|---|---|
| Presidente: | Prof. Dra. Fernanda Maria Barquinha Tavares Vieira Barbosa |
| Arguente: | Prof. Dr. João Paulo de Sousa Ferreira Fernandes |
| Vogal: | Prof. Dr. Jácome Miguel Costa da Cunha |


FACULDADE DE
CIÊNCIAS E TECNOLOGIA
**UNIVERSIDADE NOVA** DE LISBOA

**March, 2016**

**A Structured Approach to Document Spreadsheets**



Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "unlthesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2]. [1] https://github.com/joaomlourenco/unlthesis [2] http://www.di.fct.unl.pt

*para os meus pais.*

# Acknowledgements

I would like to express my sincere thanks to my advisor Jácome Cunha for support and guiding my research and mainly, for his patience to correct my writing. I also extend my gratitude to professor Miguel Goulão (Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa) for his support doing the statistical evaluation of my empirical study.

I am also grateful to Felienne Hermans and Bennett Kankuzi for their help; To Felienne for clarifying me some doubts and providing a link to a parser that I have used in my framework; To Bennett for giving me authorization to reuse in my work, the code developed in his framework during his PhD.

I also want to thank NOVA LINCS for financing my trip to Covilhã, to present my scientific paper.

I also would to thank to all participants of my empirical study.

I am also thankful to all my friends: to Rui for his friendship, for encouraging, for believing in me and mainly for going out with me at weekends; To Rafael for discussing with me some ideas of my framework and for helping me when I needed. To Luís, for our erasmus journey; To Guilherme, Leonardo, Alexandre, Gonçalo, Daniel, André, António, Diogo, Rafael, Carlos, Tiago, Mário, Marta, Albert for all funny moments that I spent with you; To Ricardo for our last trip to Prague; And finally, but not least important, to Rosa, for her help writing, for our funny moments, for believing me and mostly for being persistent and encouraging. Thank you all for supporting me in writing and mostly believing in me when I do not.

Last but not the least, a special thanks to my family. Words cannot express how grateful I am to my parents for all of the sacrifices that you have made to give me this education; I would like to thank my brother for helping and teaching me to drive the truck, it was really important for me doing other activity that I love.

I also want to thanks to every one which directly or indirectly have contributed in this journey.

# Abstract

Documentation is an important artefact of any software product. For once, user documentation is important so users can be more productive using spreadsheets, without the need to arbitrarily looking for the feature they want to perform a task. On the other hand, developer documentation allows for software professionals to more easily maintain the software.

This issue is even more relevant when dealing with software for non-professional programmers. In fact, in many cases documentation is not even available. For instance, for spreadsheets, and even considering an industrial setting, only around 30% have some kind of documentation. This makes their usage and maintenance very difficult.

There are many development systems allowing users to properly write documentation. For instance, JavaDoc is a well-known and used framework to document Java projects. Spreadsheets, the most used development environment for non-professional programmers, do not have such a system. Thus, it is important to create a documentation support mechanism, but specific for spreadsheets and their users and developers.

In this document we introduce a structured approach to document the different parts of a spreadsheet, from cells to complete workbooks. Our proposal is supported by an Excel add-in to ease the writing and reading of the documentation. Furthermore, it is also possible to consult the documentation in a web page. The add-in also allows to import and export the documentation as an XML file, to facilitate its exchangeability.

Our methodology allows developers to document, focused on (future) users, the input and output cells of a spreadsheet, but also focused on (future) developers, to technically document the formulas to facilitate their maintenance.

The empirical validation we performed shows that, in most cases, our approach improves the effectiveness and efficiency of spreadsheet users/developers.

**Keywords:** Spreadsheets, User Documentation, Developer Documentation, Tools, Excel, Add-in.

# Resumo

A documentação é um artefacto importante em qualquer software. Assim sendo, a documentação para os utilizadores é importante, pois irá aumentar a sua produtividade sem a necessidade de andar à procura arbitrariamente do que precisam para desempenhar uma tarefa na folha de cálculo. Por outro lado, a documentação específica para programadores torna mais fácil a manutenção/atualização do programa em causa.

Esta questão é ainda mais relevante quando se trata de programas para utilizadores que não são programadores profissionais. De facto, em muitos casos, essa documentação nem está disponível. Por exemplo, para folhas de cálculo, e mesmo considerando um cenário industrial, apenas cerca de 30% têm algum tipo de documentação. Isso implica que a sua utilização e manutenção se torna muito difícil.

Existem muitas ferramentas que permitem aos utilizadores escreverem corretamente documentação. Por exemplo, o JavaDoc é uma ferramenta muito conhecida e usada para documentar projetos Java.

As folhas de cálculo são o ambiente de desenvolvimento mais utilizado pelos programadores não profissionais, mas neste caso não existe um sistema semelhante ao JavaDoc. Portanto, é importante criar um mecanismo de suporte para a documentação, mas específico para utilizadores e programadores de folhas de cálculo.

No presente documento, introduzimos uma abordagem estruturada para documentar as diferentes partes da folha de cálculo, começando nas células e acabando na folha de cálculo em si. A nossa proposta é baseada num suplemento para o Excel de forma a facilitar a escrita e leitura da documentação. Além disso, é também possível consultar a documentação numa página web. O suplemento permite também importar e exportar documentação. A nossa metodologia permite aos programadores documentarem as células de entrada e de saída da folha de cálculo, focando-se nos utilizadores (futuros), mas também focando-se nos (futuros) programadores, permite documentar mais tecnicamente as fórmulas facilitando a manutenção da própria folha.

A avaliação empírica que realizamos mostra que, na maioria dos casos, a nossa abordagem aumenta a eficácia e a eficiência dos utilizadores/programadores de folhas de cálculo.

**Palavras-chave:** Folhas de cálculo, Documentação para Utilizadores, Documentação para Programadores, Ferramentas, Excel.

# Contents

# List of Figures

# List of Tables

# Acronyms

**EuSpRIG** European Spreadsheet Risk Interest Group.

**IDE** Integrated Development Environment.

**VBA** Visual Basic for Applications.

**XML** eXtensible Markup Language.

**XSLT** eXtensible Stylesheet Language Transformations.

Chapter 1

# Introduction

Spreadsheets are used by a continuously growing number of people and organizations. Indeed, spreadsheet systems are the most used programming system [15], especially by non-professional programmers, the so-called end users. As in other programming languages/environments, it is quite common to find spreadsheets with errors. Indeed, the error rate within spreadsheets can be up to 90% [18]. Hence, the European Spreadsheet Risk Interest Group (EuSpRIG)[1] regularly updates their web site with new stories reporting the losses (economical, brand recognition, etc.) caused by errors in spreadsheets to companies and other entities.

Many reasons exist for this scenario: the lack of abstraction, of a testing methodology, or a (very) weak type system. Some errors can also be explained by the lack or poor documentation [21]. Indeed, in many of the cases reported by EuSpRIG, the lack of or bad documentation is mentioned. Moreover, software tends to lose some of its efficiency when no proper documentation is available [7]. Without the proper documentation users and developers have more difficulties in understanding, using, and updating the software. The same happens for spreadsheets. In a recent study in a financial institution, researchers found that 70% of users that receive spreadsheets from colleagues have difficulties understanding them [11]. This transferring scenario is quite common as 85% of the study participants reported doing so. The same authors reported that spreadsheet users browse them for hours trying to understand their content since only one third has some kind of documentation [10].

Unfortunately, spreadsheet systems do not have a proper form to document their programs. In modern spreadsheet systems it is possible to add general notes to a cell, but that is a very unstructured way of doing documentation, when compared to what tools like JavaDoc[2], for the Java programming language, allow. This can be compared to writing ad-hoc comments in a textual programming language which makes it quite hard for spreadsheet developers to actually document their spreadsheets. For instance, in [9] the authors analysed more than 15.000 spreadsheets available in the Enron Email Archive [13], containing the emails from the

[1] http://www.eusprig.org/

[2] http://www.oracle.com/technetwork/articles/java/index-jsp-135444.html

Enron oil corporation. They found that some spreadsheet documentation was in the emails themselves, instead of being in the spreadsheets. This shows that there is the need to document, but not the proper means.

Users tend to workaround this situation documenting their spreadsheets in the best way they can. Some write the documentation on a separate worksheet and reference to it informing that such worksheet is the documentation of the spreadsheet. In this case, it is not possible to see the documentation and the corresponding document artifacts at the same time, as one can do using for instance JavaDoc, making it difficult to relate the documentation with the actual spreadsheet content. Others, write the documentation on the worksheet with the content, close by the cells they want to describe. However, in these cases the users are inserting extra cells in the spreadsheet, which are not part of the program, increasing its complexity and making it more disorganized.

These kinds of documentation make users question its interest. Although it is important to document software, it cannot be done in any way. It is important to write and organize it in such a way that the target readers will get what they want. A good documentation will increase the users' efficiency and effectiveness, and thus, their productivity [7, 19]. Indeed, JavaDoc is a good example of a successful way of documenting software [14].

## 1.1 Challenges

The main challenge of this work is to provide spreadsheet end users a way to document and read that documentation in a structured way. Although users already try to document their programs, they tend to do it in a very ad-hoc way as there is little to no support to properly document spreadsheets. Therefore, we envision a system that supports end users in creating and reading documentation in a regular and structured way. However, given end user developers do not have the training professional programmers have, the methodologies and tools proposed must have the right level of complexity and features so they can actually use and appreciate them. This is indeed the biggest challenge of this work.

## 1.2 Our Solution

The goal of this work is to propose a methodology, supported by a tool, to guide spreadsheet developers and users to write and read documentation.

We will propose a methodology to write documentation, from the perspective of who is developing the spreadsheet, and another to read such documentation, from the perspective of who wants to simply use the spreadsheet. Both methodologies are supported by a tool, an Excel add-in termed SpreadsheetDoc.

SpreadsheetDoc allows developers to write documentation in a structured way as they can document each part of the spreadsheet in a different and targeted way.

On the other hand, spreadsheet users can read the documentation written in different ways, either within the spreadsheet itself, or in a web page dedicated to that purpose.

## 1.3 Contributions

We accomplished the following contributions:

- A methodology to document spreadsheets, from the perspective of their developers.
- A methodology to read documentation for existing spreadsheets, from the perspective of the people that simply want to use them.
- A canonical format to store spreadsheet documentation under an XML format so different applications can exchange documentation of spreadsheets.
- An open-source Excel add-in, termed SpreadsheetDoc[3], that supports both documentation writers and readers.
- A web page properly structured to display the documentation of a spreadsheet.
- A (national) conference paper — [3].
- In preparation a journal paper and a conference paper.

## 1.4 Thesis Organization

The rest of the document is structured as follows:

- Chapter 2 presents some terms used, an example of use of our framework and a schema of our solution and its description.
- Chapter 3 presents how users can document spreadsheets using our framework.
- Chapter 4 presents how users can read spreadsheet's documentation using our framework.
- Chapter 5 discusses implementation's details of our framework.
- Chapter 6 presents our empirical study to validate our framework.
- Chapter 7 presents the state of the art.
- Chapter 8 concludes the developed work and discusses future work.

[3] http://spreadsheetsunl.github.io/spreadsheetdoc/

Chapter 2

# Application Context of SpreadsheetDoc

Raymon did a study showing that good documentation improves mainly two factors: effectiveness and efficiency of users [19]. So, our framework intends to improve these two factors. Raymon describes that users usually ask colleagues when they do not understand the system they are working with. Thus, users usually lose efficiency and effectiveness, increasing the losses of their corporations. Our framework aims to improve the usability of users when writing documentation. For instance, it is possible to send spreadsheets to other users without the need to explain it because it is already documented. Thus, users will have less difficulties working on spreadsheets that were not created by them.

In [16] the author discusses two kinds of documentation: development documentation and user documentation. The former is about the software itself, its internal form, and is created for developers, with technical knowledge about the software and its implementation details. The latter is for the software users, possibly with no technical skills to understand documentation written for developers. We have also separated these kinds of documentation in our approach. On one hand formulas should be documented technically, that is, with enough technical detail so they can be updated by other developers. On the other hand, input and output cells should be documented for end users so they can know where to input their values and read the results.

Developers have a documentation different from the documentation of end users. Their documentation has descriptions that only developers are accustomed to use. For instance, developers describe how a formula on a cell works, that is, explain how the formula is constructed and what parameters should be used. To end users, the internals are not important, only inputs and outputs are relevant: they just want to use the spreadsheet. So, documentation for maintainers must have details that an end user may not understand.

In this chapter we introduce a few concepts of the spreadsheet realm (Section 2.1), briefly introduce our work through an example (Section 2.2), and describe the overview of our methodology and tool (Section 2.3).

## 2.1 Definitions

**Workbook/Spreadsheet** A workbook is a spreadsheet file. The term spreadsheet is often used to refer to a workbook, when in fact it refers to the computer program, such as Excel. We will use these terms interchangeably.

**Worksheet** A worksheet, or simply sheet, is a single page of a workbook, that is, one of the tabs that can be found at the bottom of the spreadsheet (in most spreadsheet systems).

**Cell** A cell is a rectangular box in a worksheet, that is, the intersection point of a vertical line (column) and a horizontal line (row). Its name is the concatenation of its coordinates: a letter for the column and a number for the row. It can has content, which can be plain values (for instance, 4 or Bid), or formulas (for instance, =SUM(A1:B3)).

**Row** Refers to all the cells contained in a horizontal line (given by a number).

**Column** Refers to all the cells contained in a vertical line (given by a letter).

**Range** A range is a rectangular selection of cells, containing one or more cells.

**Input Cell** An input cell is a cell that is referenced by others, but does not reference any other cell.

**Output Cell** An output cell is a cell that references other cells, but its not referenced by any other.

## 2.2 Example

In this section we describe a spreadsheet which we use as a running example. This spreadsheet, shown in Figure 2.1, was introduced in a book describing how to create spreadsheets [17].

This spreadsheet calculates the probability of winning an auction, according to a set of assumptions. Although this spreadsheet is well organized and rather small, it is already difficult to understand. In Figure 2.2 we show the same spreadsheet, but now with the formulas visible.

In fact, and since this is a well designed spreadsheet, some cells even have comments on them (denoted by the small triangle appearing in the top right corner of the corresponding cells). We list next the comments from the spreadsheet:

**F4** Decision: Bid (in $million).

**I4** Net value if the salvage value is low.

**I5** Net value if the salvage value is high.

**I8** Net value if the bid is successful.

**I9** Net value if the bid is unsuccessful.

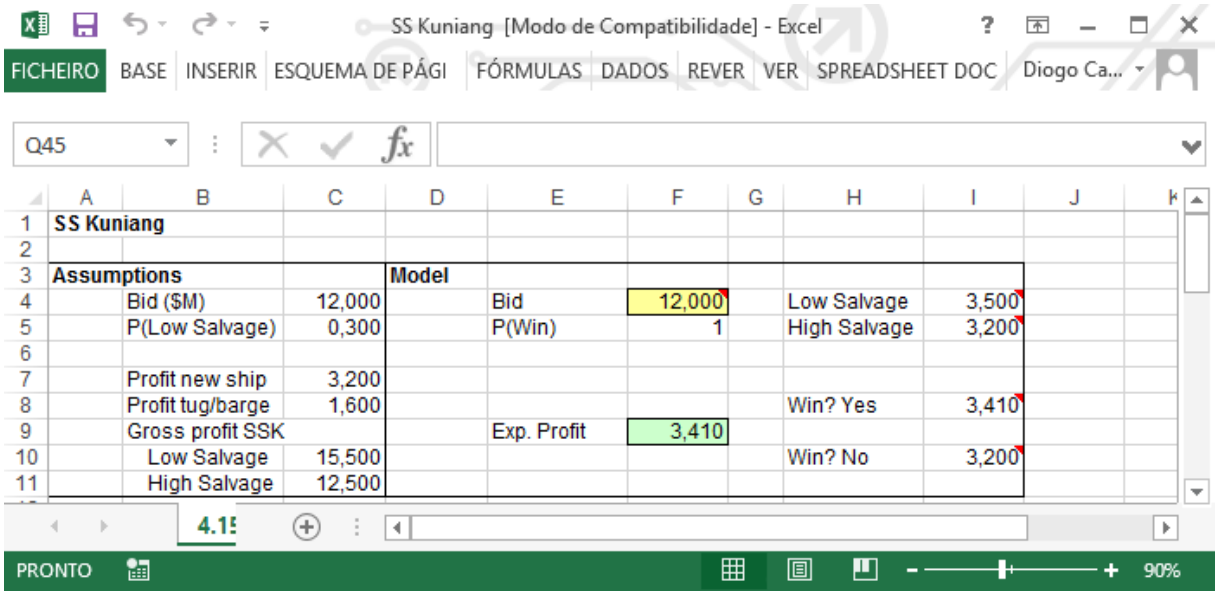



Figure 2.1: A spreadsheet to calculate the winning probabilities of an auction.

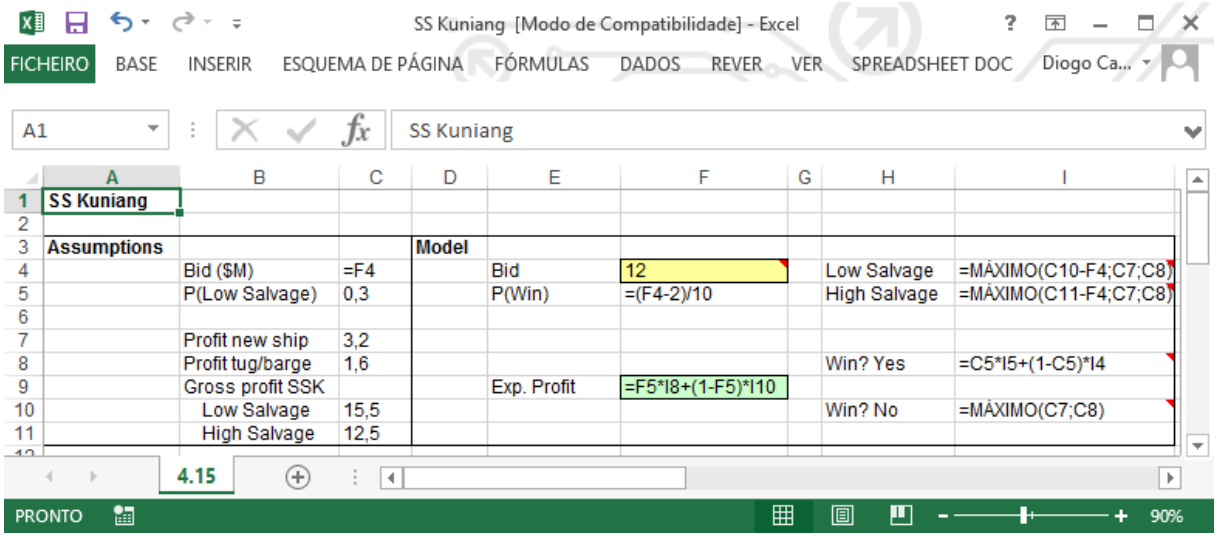



Figure 2.2: Spreadsheet of Figure 2.1 with the formulas visible.

In [17] one can read some more details about this spreadsheet and corresponding computations. For instance, cell F5 calculates the probability of winning the auction. The formula present in the book is `P(Win) = (Bid - 2)/10, for 2 ≤ Bid ≤ 12`. This formula is more direct then the one presented in the spreadsheet, and the range of `Bid` (F4) is now clear. This should be part of that cell's documentation.

Using our approach, to document such formula, the user would click on the button to describe a cell ("Cell"), under the group "Content Documentation", and the wizard shown in Figure 2.3 would appear.

Since we are documenting a cell, as in any other programming language, the developer should describe the computation, the input, and the output. The first text box allows the user to write a general description of the formula. This is similar to what a Java programmer starts

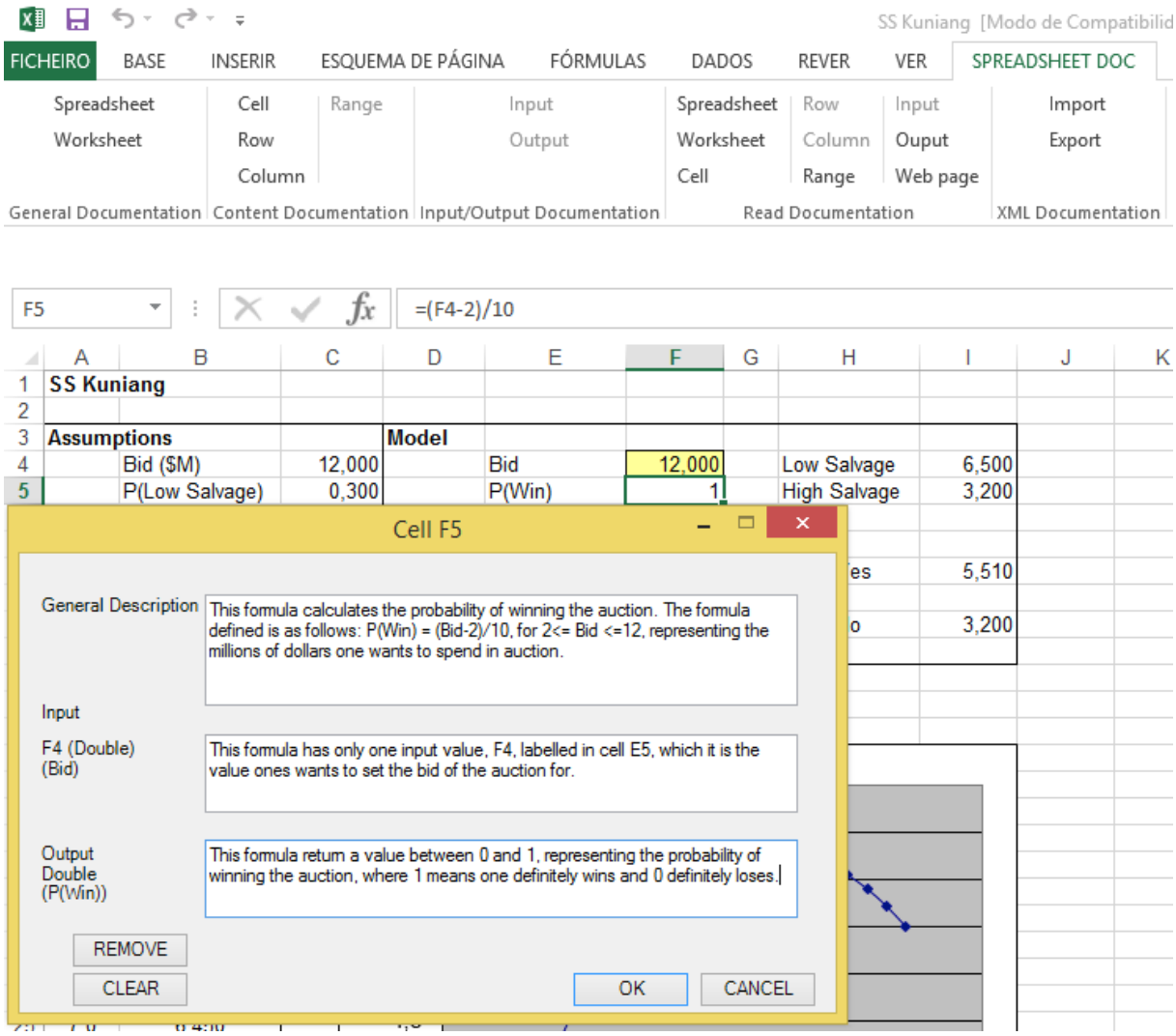



Figure 2.3: Dialogue to document a formula cell.

to write when documenting a method. Next, the user can describe the argument(s) of such formula. In this case, the input is cell F4, which is the label of the text box. Furthermore, the tool also shows the type and the header of the inputs, in this case, *double* as type and "Bid" as header. This may also help the user to detect incorrect usage of cells. Finally, the user can describe the output of the formula, again annotated with the corresponding type and label. The interested user can then read the documentation created by clicking the button "Cell" in the "Read Documentation" section of the tool. The documentation can also be read in a web page, as we will explain in Section 4.2.9.

## 2.3 Overview of Our Approach

Our framework is structured in three parts. The first part is where the user writes, reads, and updates all the documentation. This is done using the corresponding buttons listed in the ribbon, as illustrated in Figure 2.3. SpreadsheetDoc is composed of five different groups of functionalities: General Documentation, Content Documentation, Input/Output Documentation,

Read Documentation, and XML documentation.

The second part of our framework is the possibility of importing and exporting eXtensible Markup Language (XML) files with the documentation. In fact, the XML file is responsible for having all the spreadsheet's documentation. So, it is important to allow users to manipulate it, that is, replace the existent documentation with new. When a developer updates the documentation of one spreadsheet, all changes are local. So, users are working on an outdated spreadsheet until the developer publishes the new version of the documentation using the XML file. Thus, it is important the possibility of exporting updated documentation.

Such XML file can be used in different ways. For once, it is used by the tool itself to create a web page where the user can read the spreadsheet documentation, possibly with links to documentation of other spreadsheets, if they are referenced.

When exported, this XML file can also be used by other tools as they wish. For instance, it can be used by other Excel add-ins to show the documentation in a different way, or by add-ins for other spreadsheet system such as LibreOffice or OpenOffice so they can open Excel spreadsheets, but also their documentation.

It is also possible to read an XML file to import documentation written in other tools. This makes it easier to exchange spreadsheet documentation. For instance, it allows the user to import a new version of the documentation the developer may have written.

With this file, users have the possibility of having consistent documentation (importing) and also the opportunity of updating it and sending to other users (exporting).

Our framework has two buttons: "Import" and "Export". Both buttons open a generic dialogue box, one to open a file (import) and another to save a file (export).

When users are importing documentation they have to select an XML file. If they are exporting documentation, they have to select the folder where the XML file will be saved.

The XML file must follow a certain format given by the schema presented in Appendix A.

The third part of the framework is where the user reads the documentation on a web page. This web page is generated based on the XML file already created. Such web page can potentially be consulted by other people. For instance, inside an organization there can be a server with all the web pages of all spreadsheets available, and users can search for one spreadsheet's documentation, helping them to implement a new needed functionality.

Figure 2.4 illustrates the potential interactions users can have with the environment SpreadsheetDoc creates.

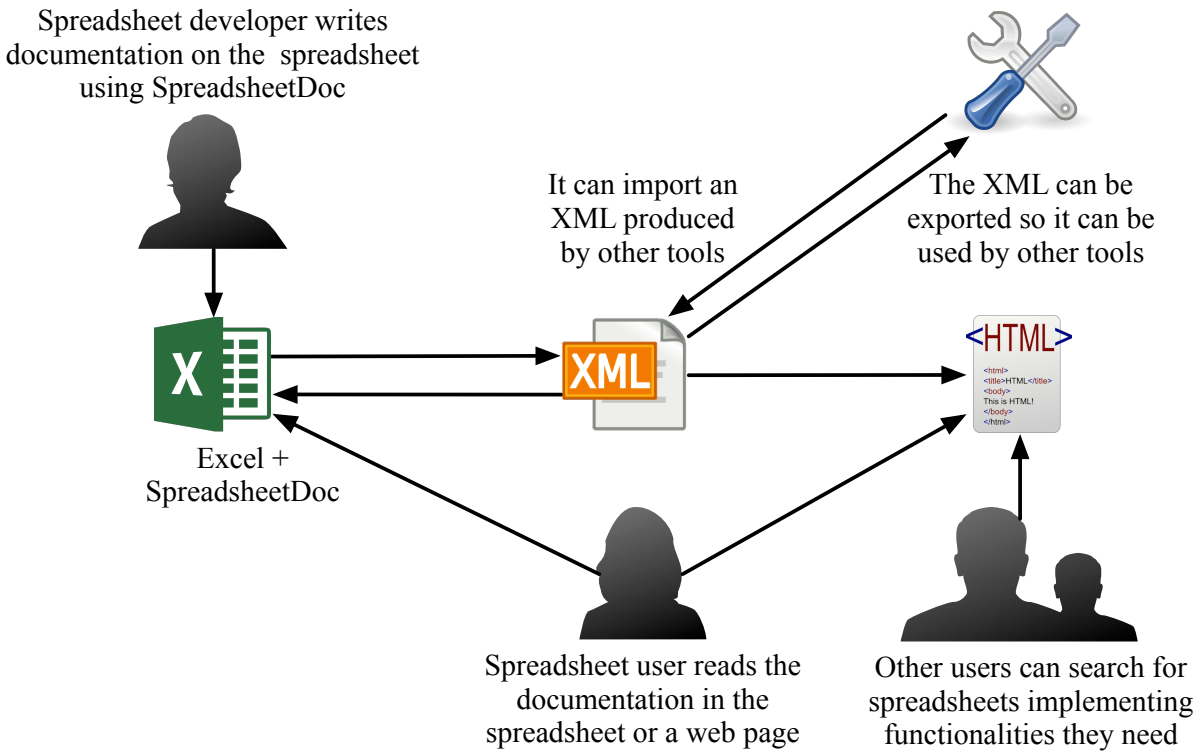



Figure 2.4: The possible interactions with the SpreadsheetDoc environment.

Chapter 3

# Documenting a Spreadsheet Program

In this chapter we explain in detail how spreadsheet developers should document their spreadsheets. In Section 3.1 we start by generically explain how to document a spreadsheet, and in Section 3.2 we introduce the SpreadsheetDoc functionalities and how they should be used to document a spreadsheet program.

## 3.1 Proposal to Document Spreadsheets in a Structured Way

It is important to document a spreadsheet in a structured way for end users to understand their purpose. Thus, to document a spreadsheet, developers should follow the next steps:

1. When developers create a spreadsheet they should start by describing its purpose in order to give users an overview of the document.

2. Once a new worksheet is added, developers should document before adding modifications to it.

3. When a cell, input, output, range, row, or column is relevant they should document it as a way to simplify user's comprehension of the spreadsheet and it should be done during the development phase.

4. When developers perform modifications on documented cells, they should review the content in order to update its documentation.

When comparing this structured way of documenting a spreadsheet with a very used framework, such as JavaDoc, it is possible to conclude that both of them are comparable. The first part is similar to documenting a Java program and each class and the second part is identical to comment a Java method. This, or any other methodology for documenting spreadsheets, becomes difficult to follow without the proper tool support. Thus, in the next section we introduce the features SpreadsheetDoc provides to document a spreadsheet.

## 3.2 Using SpreadsheetDoc to Document a Spreadsheet

In this section we are going to describe how to document each part of a spreadsheet using SpreadsheetDoc.

Taking in consideration the spreadsheet developer, we have created three distinctive groups, each one with different developers' functionalities: General Documentation, Content Documentation and Input/Output Documentation (this can be seen in Figure 3.1). The General Documentation group is composed of two functionalities: documenting a spreadsheet and each worksheets aiming to improve users understanding of the spreadsheet behaviour, namely the spreadsheet purpose (trough the spreadsheet button) and how each of its sections works (trough the worksheet button).

The second group - Content Documentation - comprises four functionalities: documenting cells, rows, columns, and ranges. Therefore, the documentation generated by this group focus on users understanding of mentioned contents.

The group regarding Input/Output Documentation has the purpose of documenting input and output cells. In this group, developers should mark all input and output cells along with a textual description. Whenever a cell is signed as input, it cannot be signed as output at the same time due to their distinct purpose. Hereupon, the documentation generated by this group acts as a guideline for users. Therefore, users are able to immediately identify where to input the data and where to read the results.

In the following subsections we explain in detail the objective of each group of functionalities.

### 3.2.1 Documenting a Spreadsheet

Documenting a spreadsheet functionality (segment of the General Documentation group) allows to document spreadsheet document as a whole. Overall, an organization makes use of numerous spreadsheets. For instance, the oil company Enron exchanged more than 15.000 spreadsheets between their emails[9]. Taking this example in consideration, it is important to document the spreadsheet file as a way for users to distinguish each file and possibly reuse them. Therefore, when (firstly) documenting the spreadsheet, the developer should write a general description of the spreadsheet itself.

When developers create a new spreadsheet they should start by clicking on the "Spreadsheet" button. Hence, this button opens a dialogue box with a text box inside where developers can write the spreadsheet's general purpose. In this dialogue three buttons are shown: `clear`, `ok`, and `cancel`. The `clear` button, as the name suggests, clears the text box. The `ok` button saves the dialogue box state. Finally, the `cancel` button drops all changes inside the dialogue box. We show in Figure 3.1 the wizard shown by our tool for our running example (Section 2.2), and the description we added.

After writing and saving the documentation, if the user clicks again in the button, the same wizard will be shown, but this time showing the recorded text.

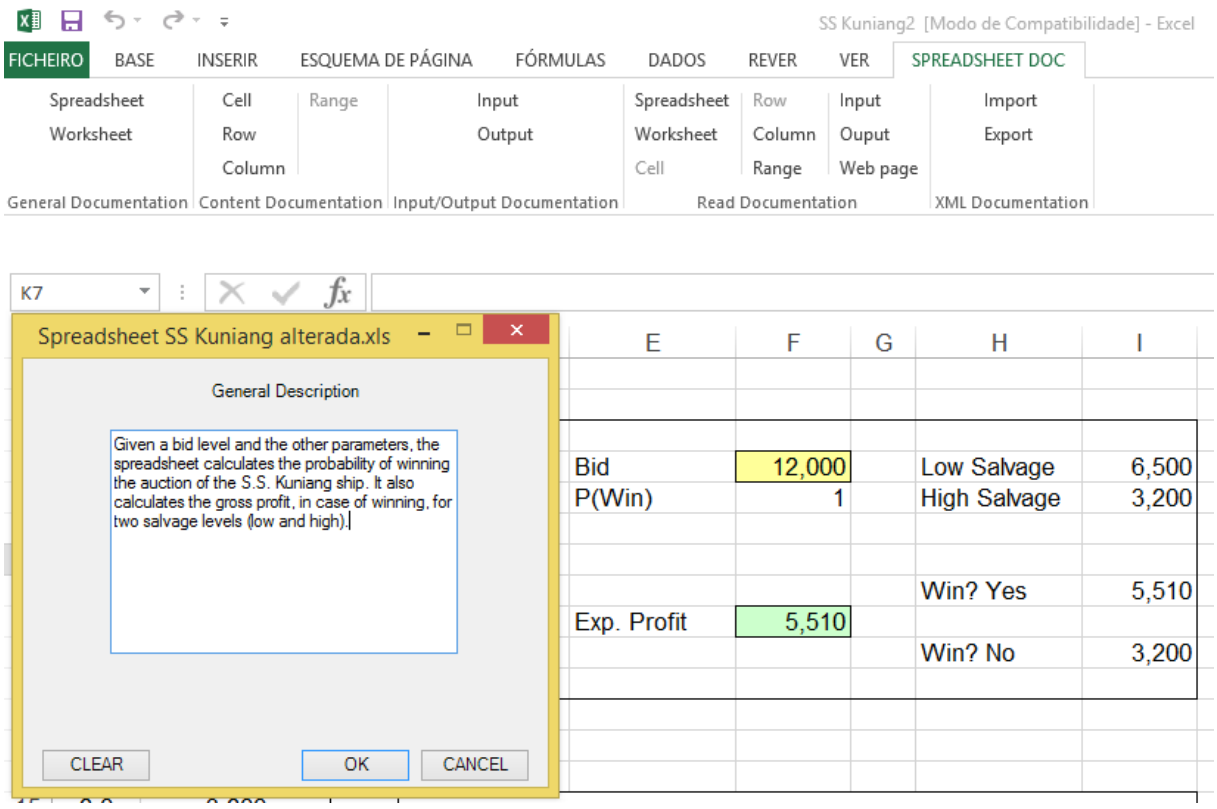



Figure 3.1: The "Spreadsheet" button wizard to document a spreadsheet file.

### 3.2.2 Documenting Each Worksheet

A spreadsheet can have dozens, even hundreds of worksheets. For the EUSES spreadsheet corpus the biggest spreadsheet has 106 worksheets [6], and for the Enron corpus, 175 worksheets were found in a single spreadsheet file [9]. Thus, it is quite important to document each of these worksheets, otherwise it becomes impossible to know what each one is doing.

The "Worksheet" button, from the General Documentation group, has a structure similar to the previous button, but in this case users should document the behaviour of the worksheet and not of the spreadsheet. Inside each worksheet the user should click this button and write the corresponding documentation. It will be associated with the worksheet the user is in.

The wizard shown is similar to the one in Figure 3.1, and thus we omit it.

### 3.2.3 Documenting a Cell

A spreadsheet can have thousands of cells. Indeed, the finest grain in a spreadsheet structure is a cell. Although in most cases it is not necessary to document each cell individually, some of them must be documented so one can understand how the spreadsheet works. The developer must decide which ones deserve to be documented.

For the EUSES spreadsheet corpus the biggest spreadsheet has 889.952 cells [6], and for the Enron 113.134 cells [9]. With our framework it is allowed to document each cell individually. So, in our tool there are two ways of documenting a cell: documenting a cell containing a value or document a cell containing a formula. The dialogue box shown is different as it is necessary

to document different things.

The "Cell" functionality, from the group Content Documentation, can be used to document each and every cell.

If the cell is a plain value (e.g. a string, a number), then the wizard shown is similar to the one presented in Figure 3.2, showing one more button than the one presented in Figure 3.1. In fact, the dialogue box is similar. The new button, remove, allows the user to remove the current cell from the list of documented cells. If the current cell is not documented, the button is disabled, otherwise it is enabled.

The documentation writer can then describe the cell content. If the content is a formula, then the description must be more technical, so it can be updated by other developers.

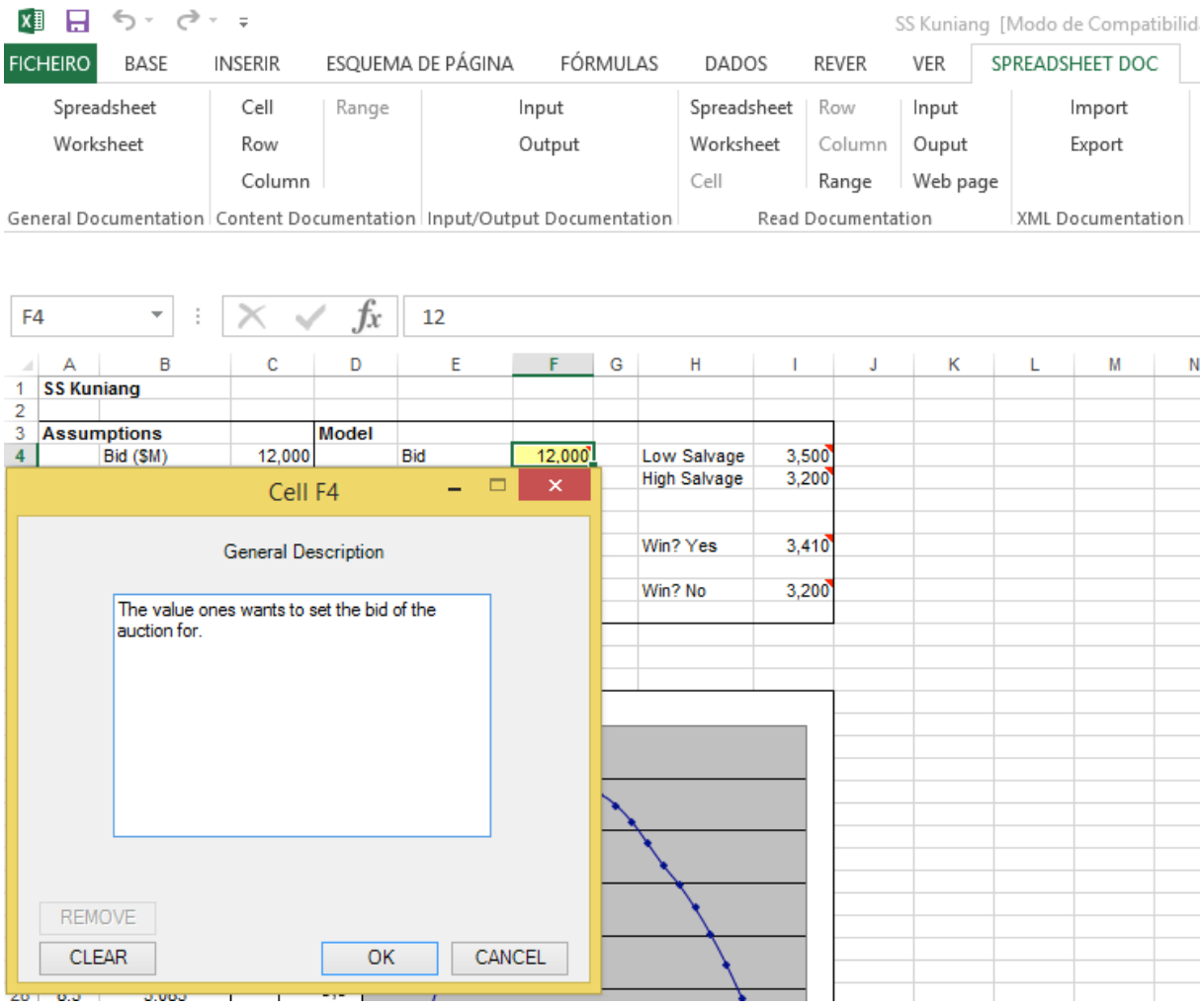



Figure 3.2: The "Cell" button wizard to document a cell with a plain value.

If the cell content is a formula, at least three different text areas must be filled in. The first text box is for the user to write a small description of the selected cell. The next text boxes are used to describe the input and output.

For each input, that is, for each reference or range, a text box is presented so the user can describe such input. Each text box has a label on the left showing two possible options. For each argument, if it is a cell range, such range is shown, so the user knows which cells he/she is

describing. The range type is also shown.

The type of a range however must be computed as Excel does not have such information. If all the cells have the same type, then such type is presented. Otherwise, we compute the type represented in more cells and present that type.

If the input is a reference to a single cell, then the tool presents its name (reference) and type as given by Excel. This can be seen in Figure 3.3.

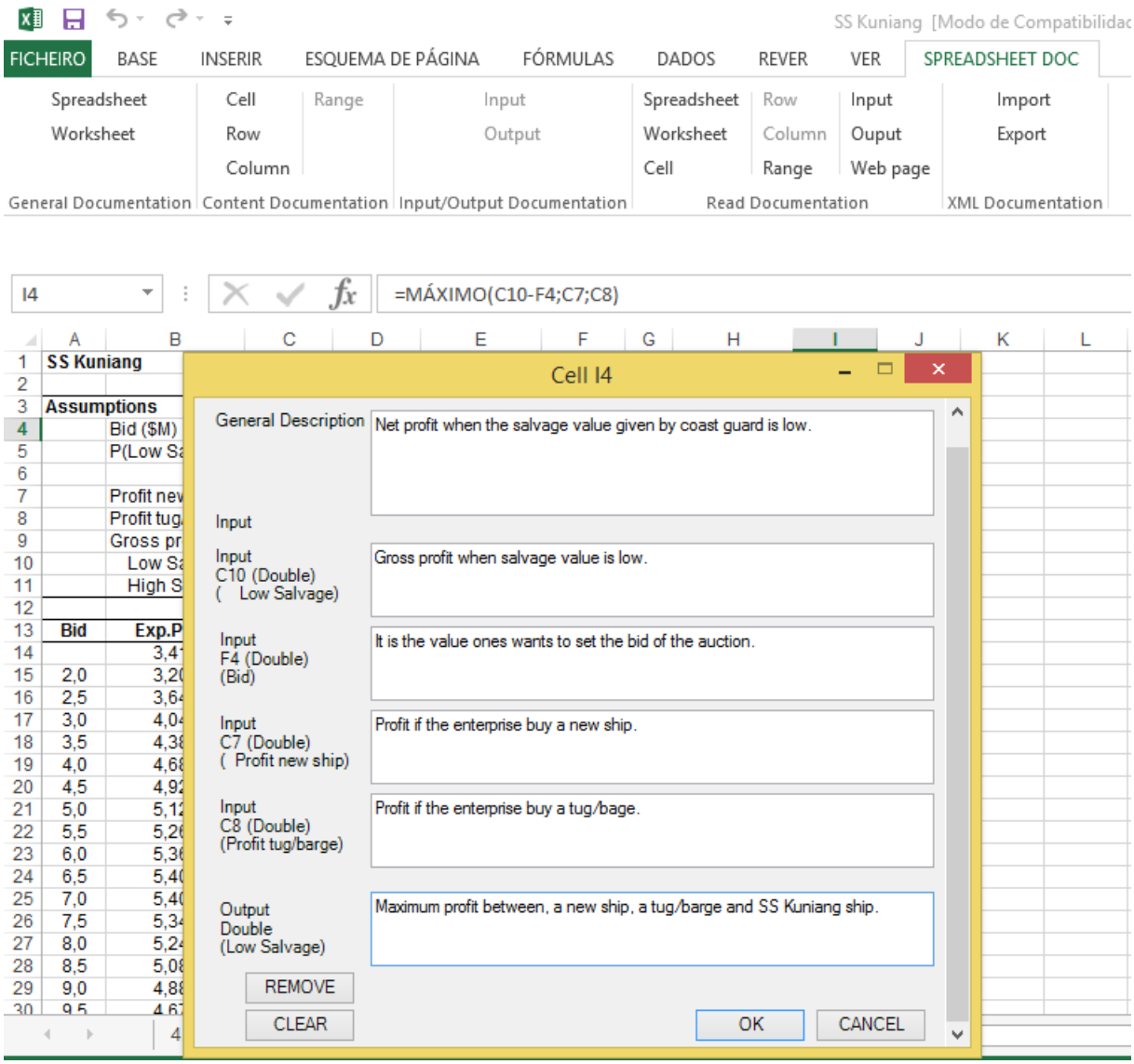



Figure 3.3: The "Cell" button wizard to document a cell containing a formula with several inputs.

Finally, the last text box is to describe the output generated by the cell. Its label is the type of the cell.

We also show the header units of each cell through the integration of an automatically inference method [12]. Hence, some documentation is given to users automatically. For instance, for our running example (presented on Chapter 2), we show the input of cell F5 as being "Bid ´", and not as being only cell F4 (as "Bid" it is the label of cell E4), since the label is much more informative than the cell reference, as shown on Figure 2.3.

As shown in Figure 2.3, the dialogue box has four buttons, already described. This documentation process can be compared to the JavaDoc tool where users document their methods. In this case, we document formulas. With JavaDoc the user writes a general description of the method, our first box, describes each method argument, our following boxes, and finally describes the return of the method, our last box.

### 3.2.4 Documenting a Row

Usually, spreadsheets users tend to develop tables containing rows and columns. For now we are going to focus on rows. Depending on the spreadsheet structure, commenting a row can be useful. Some rows are crucial for the produced results. For instance, a user can document a row describing all the row behaviour, describing in a general form the parameters that one is supposed to input on each cell of such a row. This functionality can also be used when each row of the worksheet has a particular meaning, for instance, each row in a budget spreadsheet may represent the budget for each particular entry.

The use of this functionality makes sense only if the worksheet does not have tables side by side. Otherwise, commenting an entire row can be confusing.

The "Row" button opens a dialogue box with a text box inside, similar to the one presented in Figure 3.1 (thus we omit its illustration). This button is also part of the Content Documentation group.

### 3.2.5 Documenting a Column

Previous we have described the documentation of a row. The dual applies for columns. In some cases spreadsheets are developed row oriented, but in some other cases, column oriented. Thus, the documentation writer must choose the necessary and correct functionality so the user can get the most out of the documentation.

### 3.2.6 Documenting a Range

Spreadsheets are a development framework where users have quite freedom. So, it is possible (and actually quite common) to have more than one table on the same worksheet. Then, tables have different objectives and it is important to understand what is the purpose of each one. So, we have created a functionality where it is possible to document a selected range of cells, that is, a rectangular selection of cells.

The "Range" button opens a dialogue box with two text boxes inside. Besides the dialogue boxes, it has four buttons and one checkbox, as illustrated in Figure 3.4. The first text box is in read-only mode and shows a list of all range's cells and corresponding type, as given by Excel (for instance, Double). The second text box, is for users to document the selected range. After the user adds a new range, the list of documented ranges is updated. The four buttons were already explained previously, so we are going to explain only the purpose of the checkbox. When a user clicks on it, it will surround the range with a thick black border.

This button is the last of the Content Documentation group.

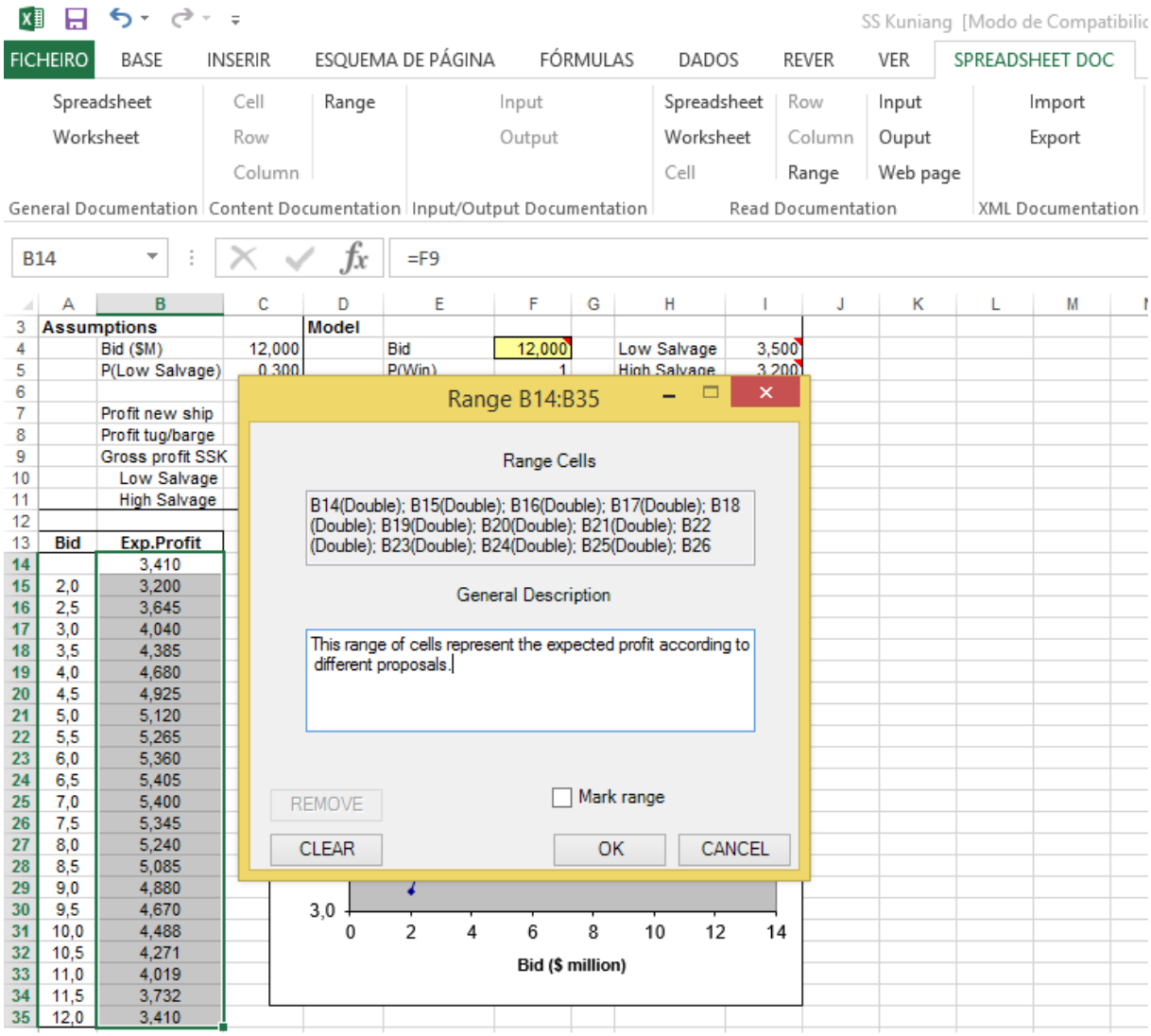



Figure 3.4: The "Range" button wizard to document a range of cells.

### 3.2.7 Documenting an Input Cell

When an end user opens a spreadsheet it is important to understand how it works. In fact, many users are not the developers of the spreadsheet.

So, more than developers, end users are the main beneficiaries of this documentation. In many cases, the user just wants to know where to input the values and where to collect the results.

The "Input" button opens a dialogue with two text boxes inside as shown in Figure 3.5. The first text box is in read-only mode and shows a list of all input cells and corresponding types. After the user adds a new input cell, the list of documented inputs is updated. The second text box allows the user to freely document the cell.

A cell is added to the list of documented inputs after the user clicks `ok`. In such a dialogue four more buttons are shown: `clear`, `ok`, `cancel`, and `remove`.

All these buttons act as described before. The "Input" button is a part of Input/Output Documentation group.

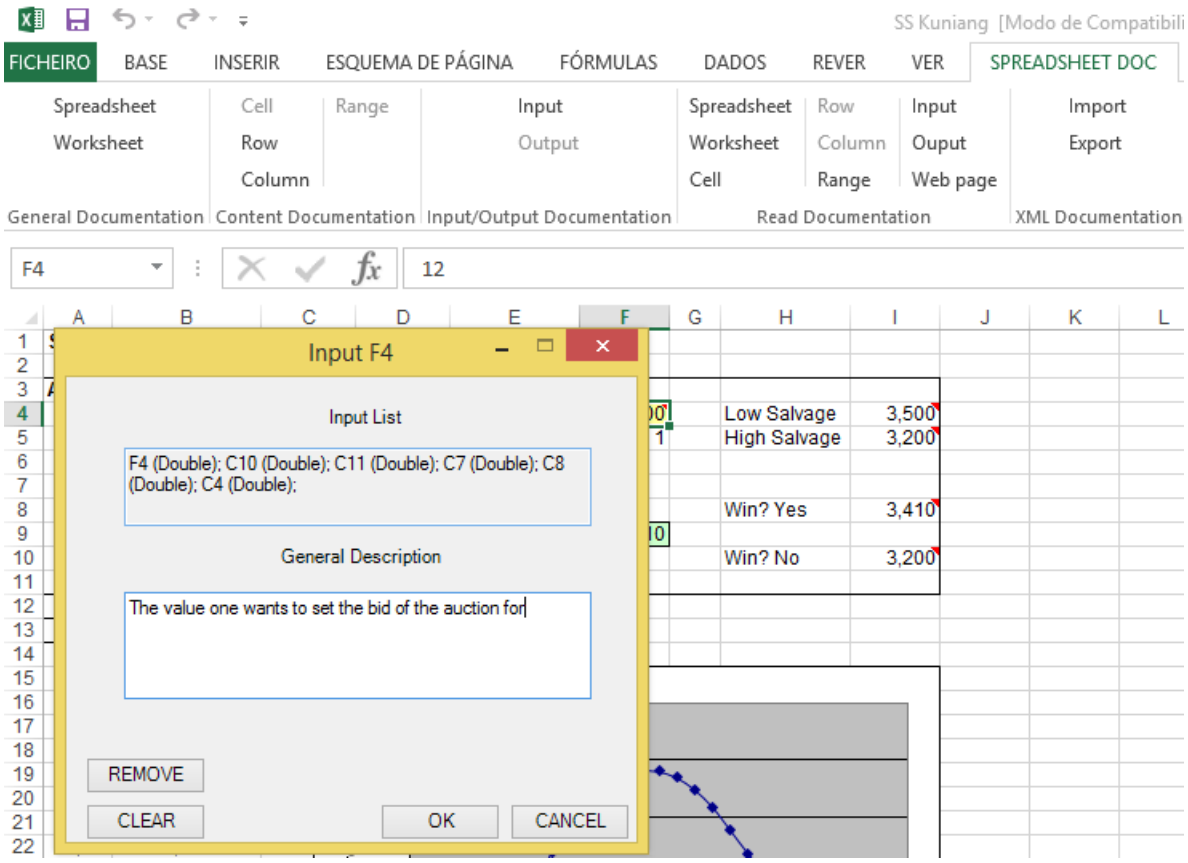



Figure 3.5: Dialogue to document an input cell.

### 3.2.8 Documenting an Output Cell

Output cells are where the user usually sees the results produced by the spreadsheet program. Thus, these are probably the most important cells for end users.

Similar to “Input”, the “Output” button opens a dialogue with two text boxes inside, now showing the list of output cells, and the place to describe the current selected output cell. For each output cell, it is also shown its type.

Chapter 4

# Understanding a Spreadsheet Program

In this chapter we discuss how users can read spreadsheets' documentation to better understanding a spreadsheet program. In Section 4.1 we explain how users should use our framework to read the existent documentation, and in Section 4.2 we detail the implemented features to read the documentation of a spreadsheet.

## 4.1 Proposal to Read Spreadsheets and their documentation in a Structured Way

It is important to understand how to read a spreadsheet and its documentation in the appropriate way. Thus, to a better understanding of the spreadsheet, end users should follow the next steps:

1. When users open a spreadsheet, they should start by reading its overview in order to fully understand its purpose.
2. Reading a spreadsheet as well as a worksheet should be done before changing any cell.
3. Additionally, users should read the documentation of the cells of interest, in case of its existence, as a way to certify if that is the correct cell to be altered.
4. Users can consult the documentation both in a web page or in the Excel.
    a) If users want to have an overview of the spreadsheet they should use the web page.
    b) If users want to know what does a specific cell, it is more efficient to consult it in the Excel. SpreadsheetDoc updates automatically the button's state when the selected cell changes.

It was possible to prove with our empirical study that web page and documentation in the spreadsheet have their own advantage. When users want to read the documentation of a specific

cell, it is faster read in the Excel than searching in the web page. On the other hand, when users do not know what are they looking for, it is better to use the web page instead of search in spreadsheet for documented cells one by one and read its documentation.

## 4.2 SpreadsheetDoc Features

In this section we will describe how to read each part of a spreadsheet.

Taking in consideration end users, we have created the Reading Documentation group. This group is composed by nine functionalities: read the spreadsheet documentation, worksheets, cells, rows, columns, ranges, input cells, output cells, and read all documentation in a web page.

When an end user opens a spreadsheet, he/she needs to know how it works. Usually users start by exploring the spreadsheet, trying to understand it before changing anything [8]. However, with the support of our framework, especially with the functionalities listed above, they do not need to explore the spreadsheet. In fact, they just have to read the needed documentation and start working. The documentation has the goal of giving users full support about how to work with a given spreadsheet without, for example, the need of asking for a colleague's help.

### 4.2.1 Show Spreadsheet Documentation

Spreadsheets are often given to users without more details and they are asked to perform several tasks [8]. Typically, as they were not the ones developing the spreadsheet, they may not understand it. Therefore, when an end user opens a spreadsheet it is important for he/she to understand how it works. Consequently, we have developed one specific button, named "Spreadsheet", which users can select in order to understand what does the spreadsheet do. This button has two possible states: enabled and disabled. It is enabled when developers have written the spreadsheet description, and disabled when they have not.

Additionally, this button opens a dialogue box with a text box inside and a close button. This text box is in read-only mode and users can read the text inside, where it explains the spreadsheet's general behaviour. The close button, as the name suggests, closes the opened dialogue box. In Figure 4.1 is shown the wizard of our tool.

As for other parts of the spreadsheet, the documentation is shown inside Excel itself so users can access the documentation and the spreadsheet at the same time.

### 4.2.2 Show Worksheet Documentation

A spreadsheet can have one or more worksheets. Then, it is important to understand what each one does, as each worksheet was design with a certain purpose. Therefore, when users start working on a given spreadsheet they should know exactly in which worksheet they have to execute the changes, in order to resolve the tasks that they are performing.

The "Worksheet" button, from the Read Documentation group, has a similar structure to the previous button. Each worksheet has associated the correspondent documentation. Therefore, when a user changes the worksheet and clicks on the button, it shows the documentation of the

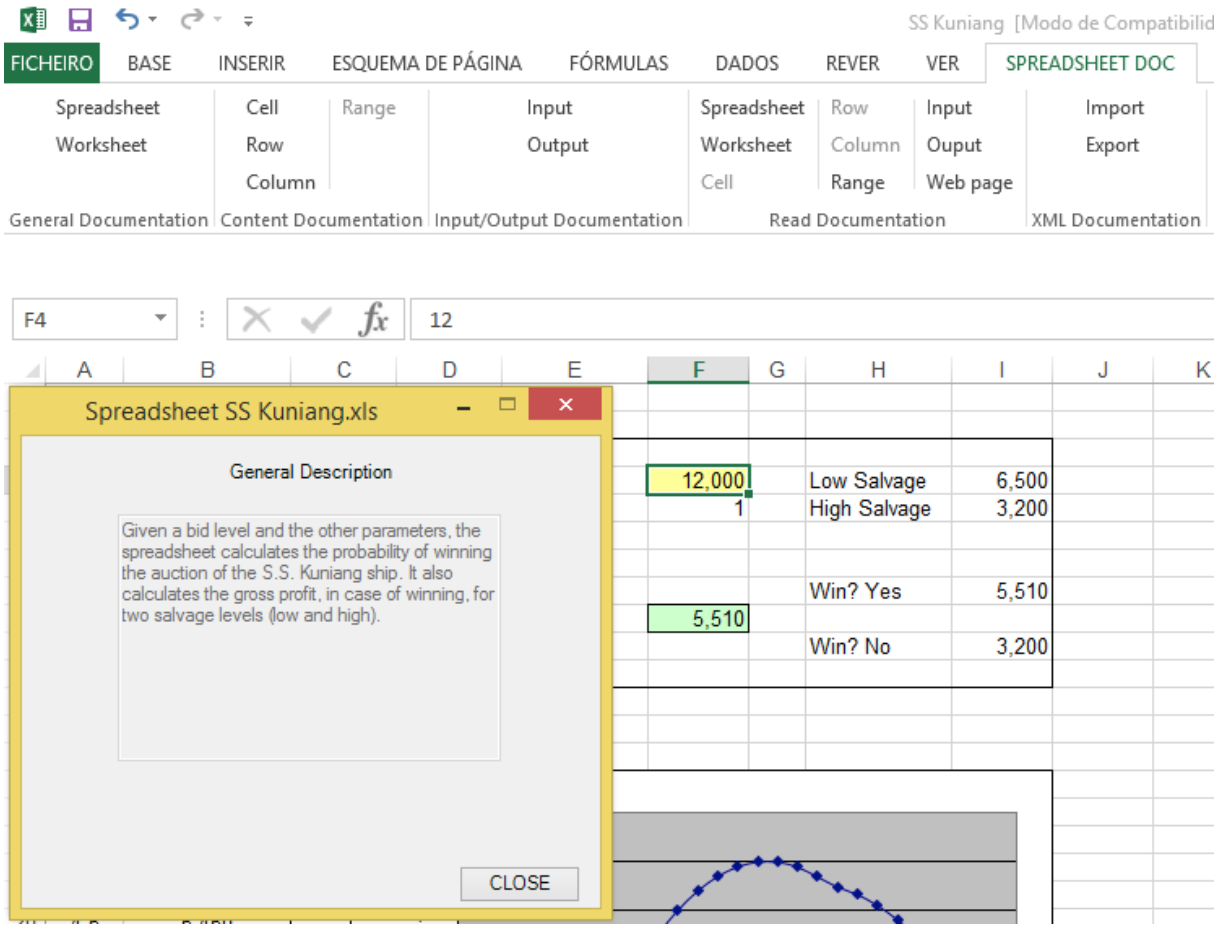



Figure 4.1: The "Spreadsheet" button wizard to read the spreadsheet documentation.

selected worksheet. As the wizard shown is similar to the one in Figure 4.1, we do not include it in this section.

### 4.2.3 Show Cell Documentation

When a user opens a spreadsheet it is usual that he/she does not know what each cell does. So, the user usually starts by selecting one cell as an attempt to understand it. Therefore, we have created the 'Cell" button as a way to simplifying the users understanding of each cell. However, some cells do not need any kind of documentation. Thus, in order to tell the user which cells have documentation, our button has two possible states: enabled and disable, where enabled means that the cell has documentation, and disabled the opposite.

The "Cell" button, from the Read Documentation group, can be used to read each and every documented cell. If the cell is a plain value, the wizard shown is similar to the one presented in Figure 4.1 (thus we do not show it in this section). However, if the content is a formula, the description is more technical in order to be updated by other developers (Figure 4.2 shows the wizard of a cell which contains a formula).

The "Cell" button opens a dialogue that is adapted to the cell content. The first text box is for the user to read the general description of the selected cell. The next text boxes are used to read the inputs' descriptions. For each input, that is, for each reference or range, a text box

is presented so the user can read each input documented separately. Each text box has a label on the left, showing two possible options. For each argument, if it is a cell range, such range is shown, so the user knows which cells he/she is reading. The range type is also shown. This can be seen in Figure 4.2. Finally, the last text box is to read the output produced by the cell. Its label is the type of the cell.

We also show the header of each cell through the integration of an automatically inference method [12], already explained.

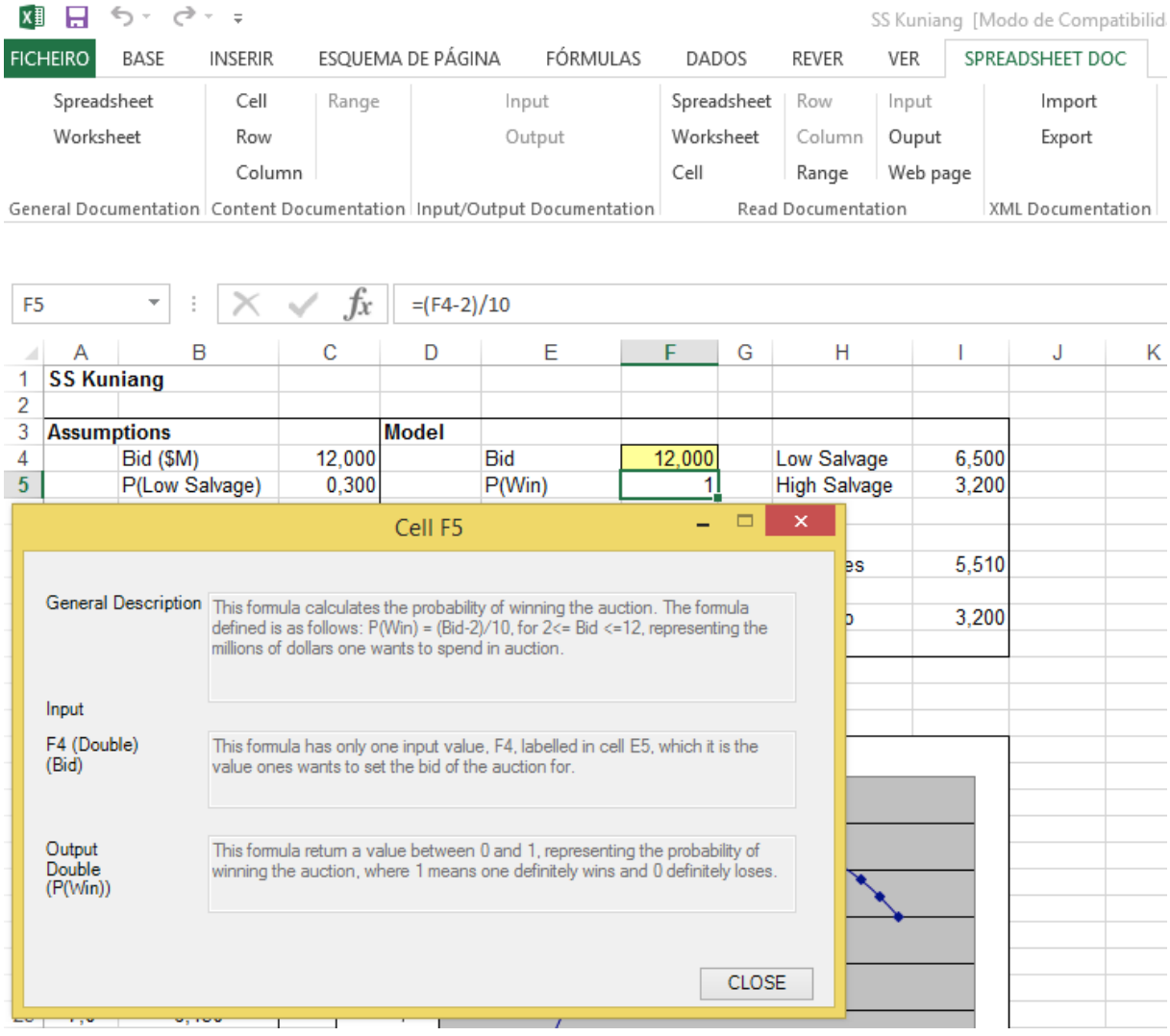



Figure 4.2: The "Cell" button wizard to read a cell with a formula value.

### 4.2.4 Show Row Documentation

Some particular rows of the spreadsheet can be important for its comprehension/use. For example, the same row on different worksheets can have different meanings. So, the "Row" button allows users to click on a row's cell to read its documentation, giving them an overview of what does each cell of the selected row.

The "Row" button opens a dialogue box with a text box (in read-only mode) inside, similar to the one presented in Figure 4.1, where users can read the row description. If the row is not

documented, then the button is disable.

### 4.2.5 Show Column Documentation

Some spreadsheets are column oriented or have important columns that should be documented. For those cases, a dual button to "Row", namely "Column" can be used to this documentation.

### 4.2.6 Show Range Documentation

Many users create several tables on the same worksheet. These tables may have different objectives and it is important, for users, to understand what each one does.

The "Range" button, from the Read Documentation group, can open two different dialogue boxes, depending on the context. It also has two possible states: enabled and disabled. Hence, when there are no ranges documented, the state is disabled. Otherwise, it is enabled. Now, we are going to explain why there are two possible dialogue boxes and when each one appears.

When users select one cell or a range of cells (the range is not documented) and click on the "Range" button, it opens a dialogue box with a text box and one close button inside. The text box is in read-only mode and shows a list of all documented ranges. In Figure 4.3 we show the wizard for this case.

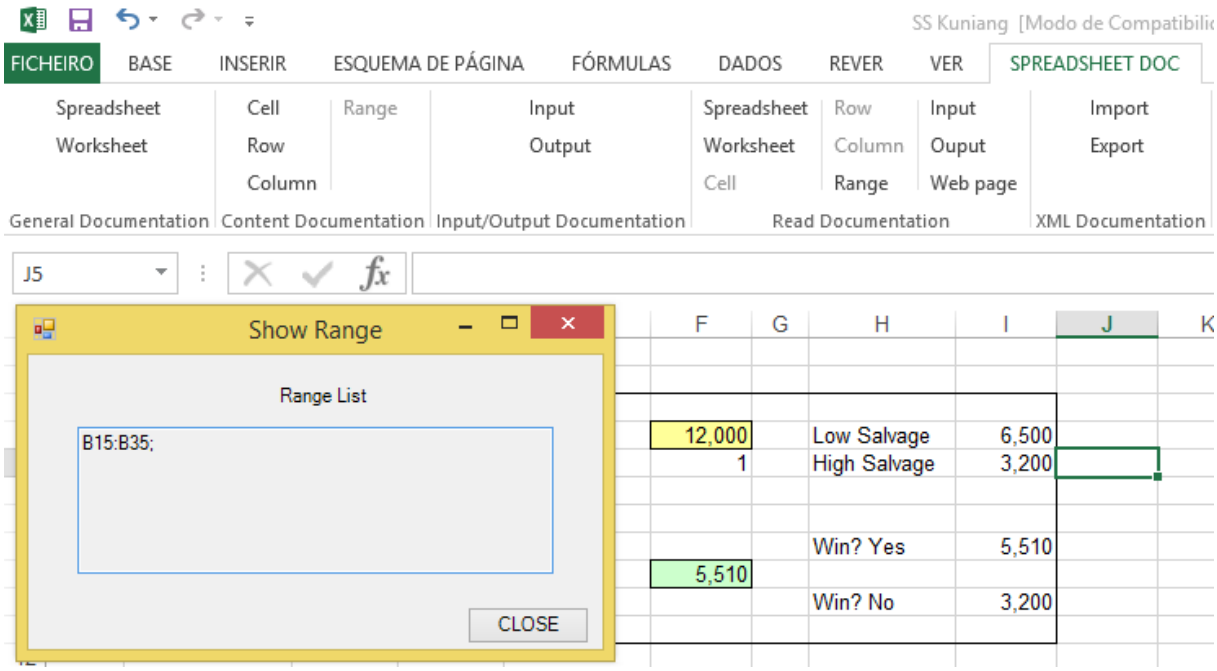



Figure 4.3: The "Range" button wizard to show existent ranges.

On the other hand, when users select an existent range and click on the "Range" button, it opens a dialogue box with three text boxes and the close button inside. All the text boxes are in read-only mode. The first text box and the button act as described before. The second text box shows a list of all the range's cells and corresponding type. Finally, the last text box allows users to read the description of the range, giving them an overview of its purpose. In Figure 4.4 it is shown the wizard of the second dialogue box described.

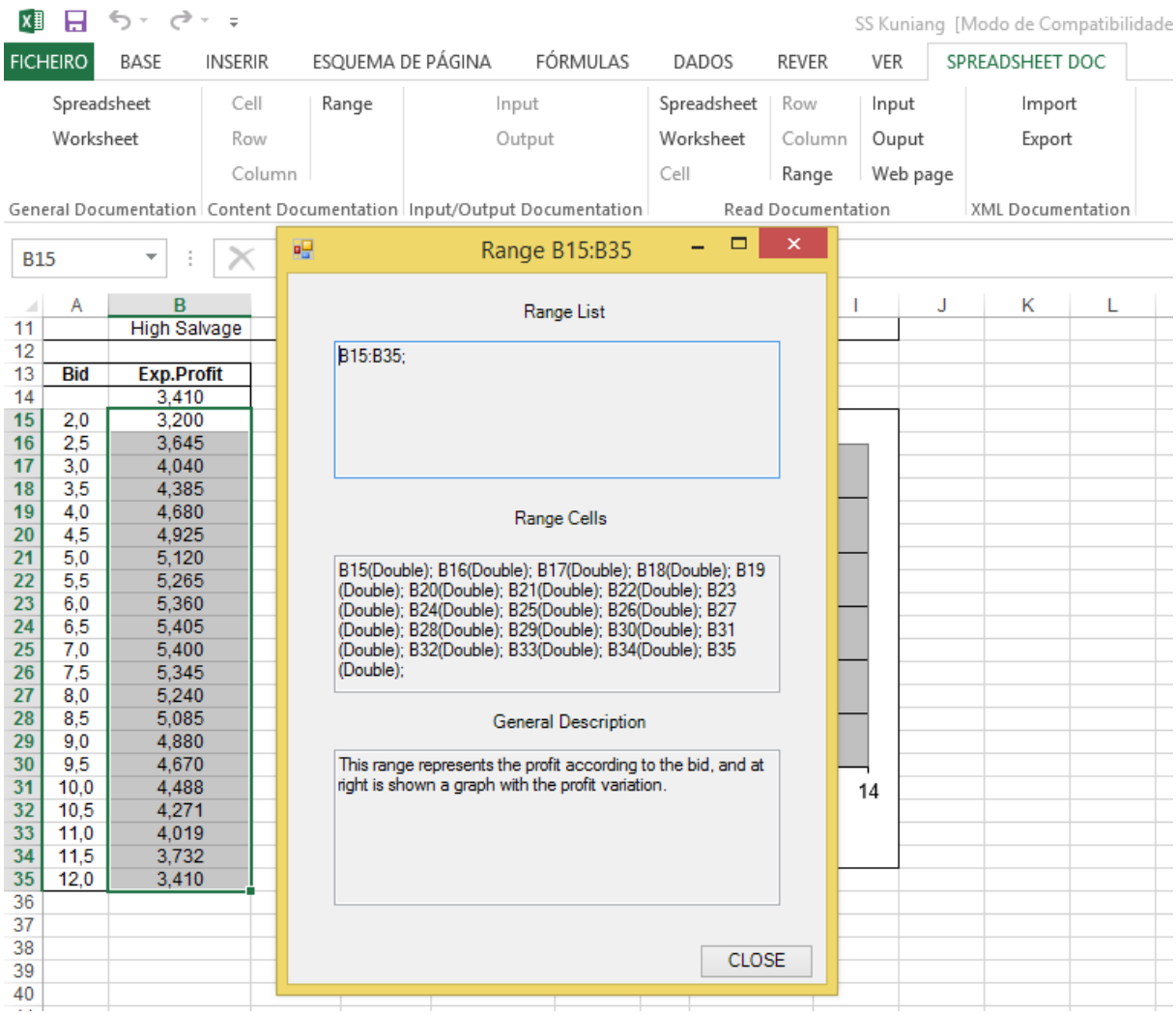



Figure 4.4: The "Range" button wizard to read the range documentation.

### 4.2.7 Show Input Cells' Documentation

When end users receive a new spreadsheet, the first thing they need to know is probably where they have to input values (inputs) and where results are shown (outputs). Correctly identifying the input cells is very import for end users, since it is where they change values to compute the necessary outcomes.

The "Input" button opens a dialogue with two text boxes, two buttons (`ok` and `cancel`), and one checkbox inside. The two text boxes are in read-only mode. The first, shows a list of all input cells and corresponding type. The second, allows users to read the input description of the related cell. The checkbox has the purpose of changing the input cell's colour. When the checkbox is checked, it paints with red the cell background colour; otherwise it removes the colour. This action with the checkbox only produces results when users click `ok`, otherwise, nothing happen. As the previous described buttons, this button also have state. When there is at least one input it is enabled, otherwise it is disabled. When an user clicks on the "Input" if the selected cell was marked as input it opens a dialogue box showing the list of all inputs and also the corresponding description. Otherwise, it shows only the list of input cells.

In Figure 4.5 we show a worksheet with two colours represented. The meaning of red colour was already explained, and on following sub-section we are going to explain the blue colour (outputs).

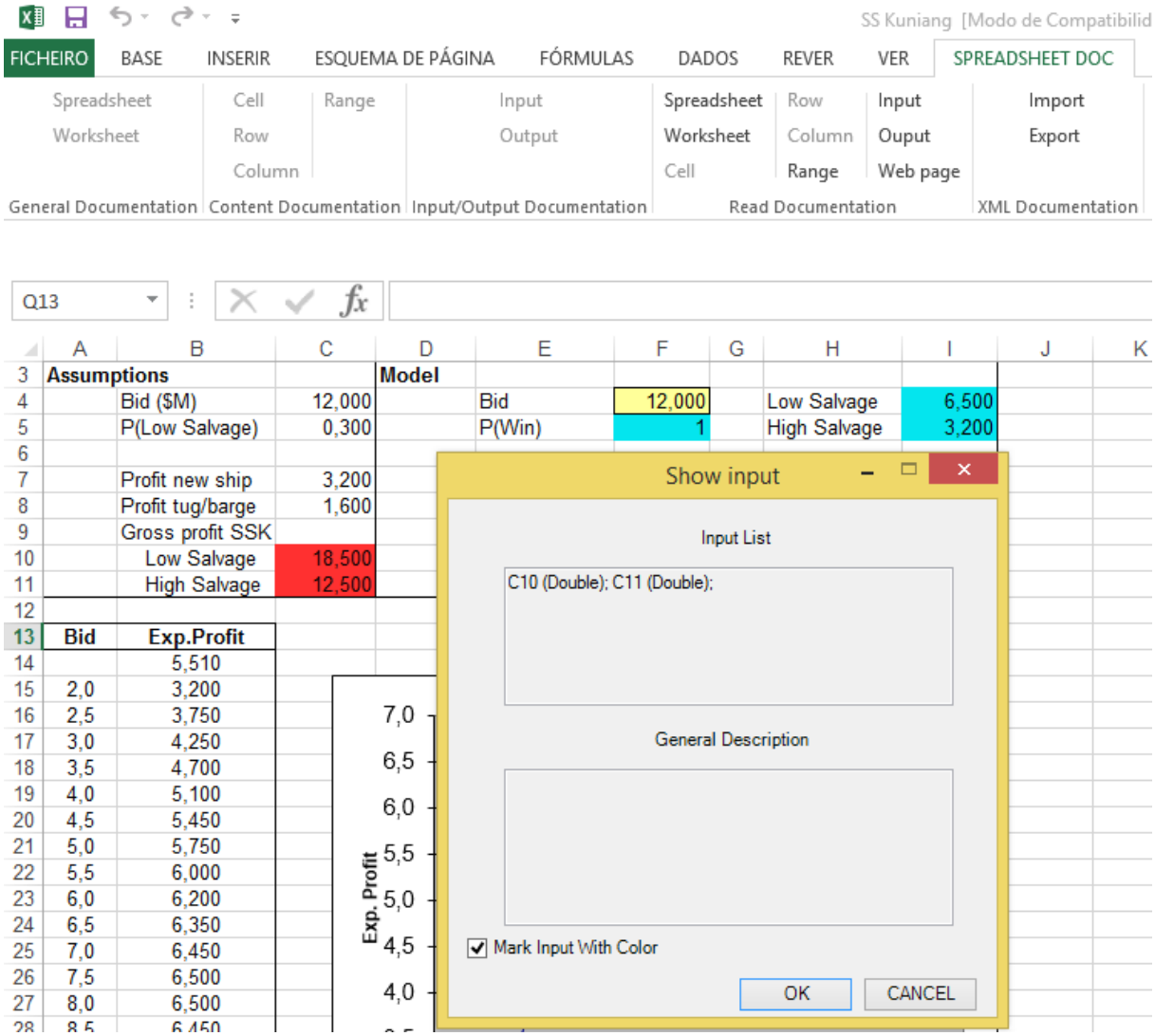



Figure 4.5: The "Input" button wizard to read the input documentation.

### 4.2.8 Show Output Cell Documentation

Every change on a worksheet's cell producing results on other cell is considered an output cell. Thus, these are probably the most important cells along with input cells for end users.

Similar to "Input", the "Output" button opens a dialogue with two text boxes (in read-only mode) inside, showing the list of output cells and the place where it is described the current selected output cell. For each output cell, it is also shown its type. It also has two buttons (`ok` and `cancel`), and one checkbox for end users productivity, acting as described before. Here, when users save the checked state, the output cells' background colour changes to blue, otherwise, the background colour is removed. Furthermore, it has two states: enabled and disabled which acts like the ones described on previous sub-section.

When an user clicks on the "Output" if the selected cell was marked as output it opens a dialogue box showing the list of all outputs and also the corresponding description. Otherwise, it shows only the list of output cells.

The opened dialogue box is similar to the one presented in Figure 4.5 (thus we omit its illustration).

### 4.2.9 Show the Complete Documentation in a Web Page

We have designed a web page in order to facilitate the interaction of end users with the complete documentation. This web page contains all the documentation of a spreadsheet, so it becomes easier for users that do not know the spreadsheet to consult all documentation together instead of searching cell by cell in the spreadsheet.

The biggest advantage of our page is the possibility to present to users all available documentation of a spreadsheet. This functionality in an enterprise context allows to push the web page to a server, providing it to all enterprise's collaborators.

For each worksheet it is presented its content, that can be composed by six sections (some may be empty if the spreadsheet developers did not write any documentation): cells, rows, columns, inputs, outputs, and ranges. Each one is represented in a tab and, when clicked, it shows its description as well as its content. Each section is represented at least by one tab.

When a cell has a formula its content is different. In this case, besides the description it has also each input and one output description, as shown in Figure 4.6.

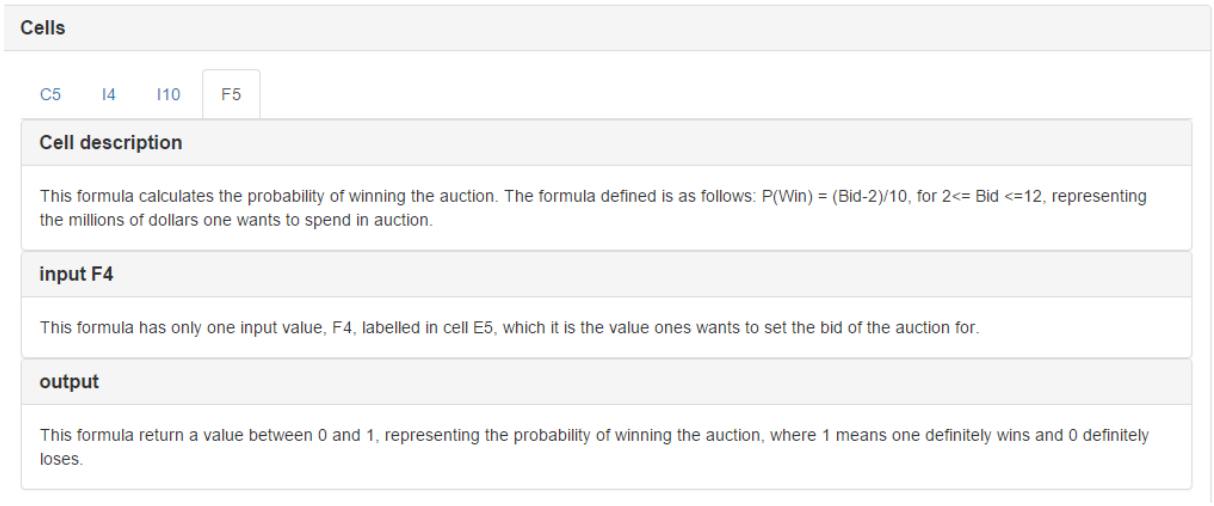



Figure 4.6: Input section in a web page when a cell has a formula.

The complete structure of the web page automatically generated by SpreadsheetDoc for the running example is shown in Figure 4.7.


Figure 4.7: Web page structure for the running example.

Chapter 5

# Developing SpreadsheetDoc

We have developed an add-in using the programming language C#, version 4.0., which was implemented using the Integrated Development Environment (IDE) Visual Studio Ultimate 2013. In order to test the add-in, we have used Excel 2013 and we have run our framework on debug mode through the IDE.

In order to develop a framework for Excel there are two possible language options: C# and Visual Basic for Applications (VBA). We have chosen the C# instead of VBA not only because C# is more used than VBA [23] but also because C# is also more powerful than VBA. As already shown on previous chapters, our framework is composed by several groups of dialogue box. All dialogue box fields are created statically with exception for cells with formula. For those, the dialogue box content is created both statically (for General Description and Output text boxes) and dynamically (for Input text boxes), which means that the number of arguments defines the number of text boxes created dynamically.

As each dialogue box has to be saved it was essential a support structure for doing it. Therefore, we have decided to use only a variable (type object) for a spreadsheet because it is only one object, and a Dictionary<Key,Value> structure where the key is each worksheet, guaranteeing that each sheet has their own cells, and the value is a list of dictionaries. Regarding the value parameter, the list of dictionaries is composed by five different dictionaries and each one follows a standard. Concerning the list entries: The first position (0) stores all saved input forms and this position contains the existent documentation for inputs; The second position (1) has the same purpose, but in this case the output forms are the ones stored; The third position (2) stores the same forms but with three distinct types of documentation. One with documentation of the own worksheet, another with row's documentation, and lastly, with column's documentation; The fourth position (3) stores all documented ranges form. Finally, the last position (4) stores forms of cells with plain value and cells with formula value. The differences between them were already explained previously. With this list, it is easier to search for desired documentation and at the same time update the buttons' state.

When users change a type of cell, for example when a cell type is double and documented,

and it is changed to string, our framework opens the dialogue box, in order to force users to update the documentation.

As the Excel does not support any file attached, when we send our spreadsheet to other user the documentation will be lost. Therefore, our documentation, which is saved as an XML file, was saved in a hidden worksheet and all changes made by developers are saved there. This allows to send one spreadsheet to other user without losing any information. Additionally, when a user opens a spreadsheet, the documentation is automatically loaded without he/she realizing it. If the worksheet does not exist, it is created without the user knowledge and all documentation is saved there. We have used an eXtensible Stylesheet Language Transformations (XSLT) file to perform transformations in order to generate a web page based on a temporary XML file, created to perform the transformation which is deleted afterwards. Thus, a web page is created following the set of rules defined by the XSLT file.

We have used two jQuery functions to change dynamically the header of each element of the web page. For example, when a user changes from cell C5 to cell C6, the header is updated. So, our implementation is extensible in the means that if someone wants to add a new functionality, he/she can add a new entry to the dictionary's list and develop all needed methods. Therefore, there is no need of doing any modifications on our code.

Our framework is available on link:
`http://spreadsheetsunl.github.io/spreadsheetdoc/`

Chapter 6

# Empirical Validation

An empirical validation is widely recognized as essential in order to validate a new framework. Therefore, we have done an empirical study, which is described in this chapter and whose results we also analyse in detail, in order to validate our framework.

Our motivation to do this study was the need to understand the differences of performance between users using the SpreadsheetDoc and users which have used traditional spreadsheet systems (Excel). In Section 6.1 we detail the design of our study. In section 6.2 we explain how we have run our study. In Section 6.3 we analyse the collected data. In Section 6.4 we interpret the obtained results and finally, in Section 6.5, we discuss the obtained results.

To execute this study we have followed others studies like [5].

## 6.1 Design

The aim of our study is to evaluate the efficiency and effectiveness of users using our framework, comparing to simply use Excel. As we have described previously, it is extremely common to find users spending considerable time amounts trying to understand and use spreadsheets. Therefore, our ambition is to mitigate this problem. Thus, evaluating the efficiency and effectiveness of users using SpreadsheetDoc was quite important.

The study we have designed was applied in an academic environment, so it was done with college students. To incentive the students of our university we have decided to raffle a voucher with the value of fifty Euro for a technology store.

We have asked to participants to perform some tasks in two distinct spreadsheets given by us. Those spreadsheets were taken from a book [17] and from Enron oil corporation database [9].

As we have described previously, we have developed functionalities to read and write documentation. In this study, we were only interested in testing the read documentation part. To create documentation we need more experienced users and they do not need much help as the users of our environment. So, with end users we can understand better if our framework is easy to use or not. Since the users that need to read the documentation are probably less experts, it

is important know if users interact easily with the spreadsheet using SpreadsheetDoc. Then, the selected subjects are similar to all non-programmer which use spreadsheets every days.

### 6.1.1 Hypotheses

The use of SpreadsheetDoc brings some advantages such as to providing users documentation about the spreadsheet and simplifying its usage. In theory, this reduces the number of errors and improves user performance. However, this needs to be tested. So, we could informally state two hypotheses:

1. In order to perform a given set of tasks, users spent less time when using SpreadsheetDoc instead of using only Excel.

2. Spreadsheets used with the support of SpreadsheetDoc have a correctness grade higher than using only Excel.

Formally, two hypotheses are being tested: $H_T$ for the time that is needed to perform a given set of tasks, and $H_C$ for the correctness grade found in different types of spreadsheets. They are respectively formulated as follows:

1. **Null hypothesis, $H_{T_0}$**: The time to perform a given set of tasks using SpreadsheetDoc is not less than that taken using only Excel. $H_{T_0}$: $\mu_d \leq 0$, where $\mu_d$ is the expected mean of the time differences.

   **Alternative hypothesis, $H_{T_1}$**: $\mu_d > 0$, that is, the time to perform a given set of tasks using SpreadsheetDoc is less than using only Excel.

   **Measures needed**: time taken to perform the tasks.

2. **Null hypothesis, $H_{C_0}$**: The correctness grade in spreadsheets when using SpreadsheetDoc is not smaller than using only Excel. $H_{C_0}$ : $\mu_d \leq 0$, where $\mu_d$ is the absolute frequency difference of the correctness grades (effectiveness).

   **Alternative hypothesis, $H_{C_0}$**: $\mu_d > 0$, that is, the correctness grade when using SpreadsheetDoc is smaller than using only Excel.

   **Measures needed:** correctness grade for each spreadsheet.

### 6.1.2 Variables

The independent variables are: for $H_T$ the time to perform the tasks, and for $H_C$ the correctness grades (effectiveness).

### 6.1.3 Subjects and Objects

Initially, we have decided that the subjects of this study must not be computer science students neither students from related areas, and secondly, that they must have some knowledge of Excel. However, as the number of participants who fulfilled this requirement was not enough to create a significant empirical validation, we have decided to expand the subjects group and admit all students, related or not to the computer science subject.

Therefore, the subjects admitted to the present study were students from Faculty of Science and Technology of Universidade NOVA de Lisboa, with a certain knowledge of Excel. In order to find the population with the desired requirements to this study, we have created a selection questionnaire (Appendix B) as a way to evaluate the student knowledge of Excel as well as the students major. Out of a total number of thirty-eight students that filled the selection questionnaire, thirty-six fulfilled the conditions we were looking for. However, only fourteen actually appeared to participate in our study. We will give more details about participants in Section 6.3. As we will show in Section 6.2, there was no statistical difference between computer science students and others.

The objects of this study were three distinct spreadsheets that will be described later in section 6.1.4. One spreadsheet was used as tutorial, explaining how participants should use our framework. Afterwards, they were asked to realize some tasks in the remaining two spreadsheets. Some spreadsheets have some documentation. So, when users used SpreadsheetDoc, we have deleted the existent documentation and presented users with ours. If users used only Excel, them the existent documentation is available for consulting. In Figure 6.1 we show the existent documentation.

C21 =H37

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **Advertising Budget Model** | | | | | | | | | |
| 2 | **SGP/KRB** | | | | | | | | | |
| 3 | **01/01/00** | | | | | | | | | |
| 4 | | | | | | | | | | |
| 5 | **PARAMETERS** | | | | | | | | | |
| 6 | | | | **Q1** | **Q2** | **Q3** | **Q4** | | | Notes |
| 7 | | **Price** | **$40,00** | | | | | | | Current price |
| 8 | | **Cost** | **$25,00** | | | | | | | Accounting |
| 9 | | **Seasonal** | | **0,9** | **1,1** | **0,8** | **1,2** | | | Data analysis |
| 10 | | **OHD rate** | **0,15** | | | | | | | Accounting |
| 11 | | **Sales Parameters** | | | | | | | | |
| 12 | | | **35** | | | | | | | Consultants |
| 13 | | | **3000** | | | | | | | |
| 14 | | **Sales Expense** | | **8000** | **8000** | **9000** | **9000** | | | Consultants |
| 15 | | **Ad Budget** | **$40 000** | | | | | | | Current budget |
| 16 | | | | | | | | | | |
| 17 | **DECISIONS** | | | | | | | **Total** | | |
| 18 | | **Ad Expenditures** | | **$10 000** | **$10 000** | **$10 000** | **$10 000** | **$40 000** | | sum |
| 19 | | | | | | | | | | |
| 20 | **OUTPUTS** | | | | | | | | | |
| 21 | | **Profit** | **$69 662** | | **Base case** | **$69 662** | | | | |
| 22 | | | | | | | | | | |
| 23 | **CALCULATIONS** | | | | | | | | | |
| 24 | | **Quarter** | | **Q1** | **Q2** | **Q3** | **Q4** | **Total** | | |
| 25 | | **Seasonal** | | 0,9 | 1,1 | 0,8 | 1,2 | | | |
| 26 | | | | | | | | | | |
| 27 | | **Units Sold** | | 3592 | 4390 | 3192 | 4789 | 15962 | | given formula |
| 28 | | **Revenue** | | 143662 | 175587 | 127700 | 191549 | 638498 | | price*units |
| 29 | | **Cost of Goods** | | 89789 | 109742 | 79812 | 119718 | 399061 | | cost*units |
| 30 | | **Gross Margin** | | 53873 | 65845 | 47887 | 71831 | 239437 | | subtraction |
| 31 | | | | | | | | | | |
| 32 | | **Sales Expense** | | 8000 | 8000 | 9000 | 9000 | 34000 | | given |
| 33 | | **Advertising** | | 10000 | 10000 | 10000 | 10000 | 40000 | | decisions |
| 34 | | **Overhead** | | 21549 | 26338 | 19155 | 28732 | 95775 | | rate*revenue |
| 35 | | **Total Fixed Cost** | | 39549 | 44338 | 38155 | 47732 | 169775 | | sum |
| 36 | | | | | | | | | | |
| 37 | | **Profit** | | 14324 | 21507 | 9732 | 24099 | 69662 | | GM -TFC |
| 38 | | **Profit Margin** | | 9,97% | 12,25% | 7,62% | 12,58% | 10,91% | | pct of revenue |
| 39 | | | | | | | | | | |

Figure 6.1: Spreadsheet containing some documentation.

### 6.1.4 Instrumentation

As we have been describing, our study was supported by three distinct spreadsheets. The spreadsheet used to perform the tutorial only has one worksheet. So, we have decided to split it in two different worksheets. One worksheet responsible for the input data and the other responsible for the results/outputs. We have decided to perform this change because it was important to explain to participants that sometimes they should work on different worksheets when performing one task.

The spreadsheet used on the tutorial was designed to calculate the probability of winning an auction of a boat depending on the bid offer and was taken from [17]. The other two spreadsheets were taken from [17] and from an enterprise database [9]. The first is about how to spend advertising money to improve the volume of sales and from now on is termed AdBudget. The second is from a gas enterprise and stores its invoice in a month and from now on is termed EnronGAS.

Participants have received a set of tasks (Appendix D) to perform. They have to insert and consult values, consult formula's arguments and update a formula.

In order to understand the participants difficulties in the study, two questionnaires were prepared: one answered before the study (pre-questionnaire) and another answered after (post-questionnaire). Before participants left the room, we have collected from each computer the two modified spreadsheets in order to obtain information about their answers later. Note that we did not wanted to evaluate the participants skills regarding the Excel program, this study's main objective was to evaluate our framework and understand if it facilitate users comprehension of spreadsheets in general.

### 6.1.5 Data Collection Procedure

We have planned several steps to run our study, with two different options: perform the task with (1) and without (2) SpreadsheetDoc help, as an attempt of comparing the efficiency and effectiveness of performing the given tasks by users. Therefore, the first option (1) includes five different phases:

1. Filling the pre-questionnaire (Appendix B);
2. Attending and performing the tutorial on SpreadsheetDoc (Appendix E);
3. Performing the set of tasks on the two given spreadsheets, with a time limit of fifteen minutes (Appendix D);
4. Filling the post-questionnaire (Appendix C);
5. Collecting all spreadsheets, questionnaires and answers.

Regarding the second option the participants did not perform the second step (2) - Attending and performing the tutorial on SpreadsheetDoc.

In steps (2) and (6) we have had a direct participation by performing the tutorial with participants and retrieving all materials used by them, respectively. All the subjects of our study were expected to perform the two sets of tasks on the respective spreadsheets. The objective

of the study was not to compare one spreadsheet against another, but instead to compare our framework against simply using Excel.

### 6.1.6 Analysis Procedure and Evaluation of Validity

The analysis of the collected data was achieved trough the comparison of the group of participants that have performed the tasks using our framework with the group of participants that have performed the tasks without our framework's help.

Since the study is composed by several tasks, we have marked each user's time, as we show on the tables in Section 6.3.1. As a way to perform the comparison we have made the average time that participants took to complete the tasks in each group and compared both of them.

To ensure the validity of the data collected, several kinds of support were planned: constant availability to clarify any doubt; the existence of the tutorial to explain how to use our framework (this is specifically for the group that preformed the tasks with documentation); and supervise the work done by the subjects in a way that do not interfere with their work. This last point consists in observe if participants are having problems and try to help them if their difficulty is related to something that does not influence the study results.

## 6.2 Execution

The study was performed in one classroom with fifteen available computers and with a total of fourteen college students, each one in the selected session which he/she wanted to participate. Initially, we have scheduled four sessions, but as we did not have sufficient participants, we have scheduled four more sessions. Despite some subjects have performed the proposed tasks with our framework and others without it, the study has been made with the same limit of time in all sessions.

Firstly, we have pre-installed our framework on the selected computers and, before each session, we have verified if the environment was correctly set.

When the subjects were already set in the classroom, we have started by introducing the purpose of the study, explaining what we have developed so far and why was their participation important.

Afterwards, the participants started filling the pre-questionnaire, with generic information about themselves (gender, age range, course, course year) as well as some questions about their previous experience with spreadsheets. As it was already mentioned, some participants have done the tasks with the documentation provided by our framework and others have performed the tasks without it, which means that they only used the existent documentation in the spreadsheet.

During the sessions where participants used our framework, we have given a tutorial and during it, we have answered all the questions that the participants had, making sure that they could use the framework correctly. Then, they have had fifteen minutes to perform all the tasks in each of the spreadsheets without our assistance (fifteen minutes per spreadsheet).

We have decided which sessions used our framework and which did not, in order to balance the number of participants with documentation and without documentation. Regarding the

sessions, we have decided to alternate the spreadsheet which the participants used to start the study. In other words, some participants have started the study with the Adbudget spreadsheet and others with the EnronGAS spreadsheet. This was quite important to get more realistic and even results, as during the tasks performed in the first spreadsheet participants are still learning how to work with our tool and because concentration levels start to decrease over a period of time, which could influence the time spent on each task.

Lastly, we have asked participants to answer the post-questionnaire in order to evaluate the confidence that they had on their performance during the study and afterwards we have collected the modified spreadsheet files, the questionnaires, the answers of each spreadsheet as well as the times to perform each task, thus we could analyse them later on.

## 6.3 Analysis

In order to perform quantitative analyses of this study, we have used all subject's results: 7 subjects for the spreadsheets that used our framework and 7 for the ones that did not used it.

### 6.3.1 Descriptive Statistics

**Subjects:** Basic information about the subjects was gathered, namely their gender, age, major, year of college, and familiarity with spreadsheets. From the fourteen subjects, nine were male and five were female. Most of them (ten) are aged between twenty and twenty-five, with three subject being over twenty-five and one less than twenty years old. The subjects come from different areas of study, and most of them are not from computer science or related areas. We have had students from:

- Master in Computer Science (3)
- Master in Electrical and Computer Engineering (1)
- Master in Mechanical Engineering (1)
- Master in Civil Engineering (2)
- Master in Conservation - Restoration (1)
- Bachelor in Applied Chemistry (1)
- Master in Biotechnology (2)
- Master in Chemical and Biochemical Engineering (1)
- PhD in Sustainable Chemistry (1)
- Master in Biochemistry for Health (1)

Furthermore, two questions about formulas were done to evaluate their knowledge:

- How do you do the sum of cells A1 through D3? - Twelve participants have answered correctly and two have done A1 plus D3, which we have considered incorrect.

- How do you do to know if cell A1 is greater than A3? - Eleven have answered correctly and three "I do not know".

Thus, most participants were familiar with basic Excel.

**Time spent:** Differences were found regarding the time that subjects used to perform the tasks. The minimum times recorded on each spreadsheet was by participants using our framework, with average times being minor than without it.

Table 6.1 and Table 6.2 show the time used for each participant to perform their tasks and an average time used for each question, on Adbudget for informatics and non-informatics **using** documentation. On the other hand, the Table 6.3 and Table 6.4 have the same goal but in this case the subjects **did not used** documentation. So, it is possible to compare each table results. Note that when a hyphen (-) appears on the table it means that the subject did not answered to the question.

Table 6.1: Time used for **non-informatics with** documentation, to perform tasks on the **Adbudget** spreadsheet.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 1 | 00:01:00 | 00:01:00 | 00:01:00 | 00:03:00 | 00:01:00 |
| | 3 | 00:01:00 | 00:01:00 | 00:01:00 | 00:03:00 | 00:03:00 |
| | 5 | 00:01:00 | 00:01:00 | 00:01:00 | 00:01:00 | 00:02:00 |
| | 25 | 00:01:00 | 00:01:00 | 00:01:00 | 00:02:00 | 00:03:00 |
| | 27 | 00:02:00 | 00:01:00 | 00:01:00 | 00:02:00 | 00:02:00 |
| Average | | 00:01:12 | 00:01:00 | 00:01:00 | 00:02:12 | 00:02:12 |

Table 6.2: Time used for **informatics with** documentation, to perform tasks on the **Adbudget** spreadsheet.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 2 | 00:01:00 | 00:01:00 | 00:01:00 | 00:03:00 | 00:01:00 |
| | 26 | 00:01:00 | 00:02:00 | 00:01:00 | 00:02:00 | 00:03:00 |
| Average | | 00:01:00 | 00:01:30 | 00:01:00 | 00:02:30 | 00:02:00 |

Table 6.5 and Table 6.6 show the time used for each participant to perform their tasks and an average time used for each question, on EnronGAS for informatics and non-informatics **using** documentation. On the other hand, the Table 6.7 and Table 6.8 have the same goal but in this case the subjects **did not used** documentation. So, it is possible to compare each table results. Note that when a hyphen (-) appears on the table it means that the subject did not answered to the question. When the sum of the times of all questions of one subjects is superior to fifteen minutes is because the subject start one task but as he/she could not solve it, he/she moved to next question finishing the previous later.

Table 6.3: Time used for **non-informatics without** documentation, to perform tasks on the **Adbudget** spreadsheet.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 10 | 00:05:00 | 00:01:00 | 00:01:00 | 00:02:00 | 00:03:00 |
| | 11 | 00:07:00 | 00:01:00 | 00:02:00 | 00:02:00 | 00:01:00 |
| | 12 | 00:03:00 | 00:02:00 | 00:02:00 | 00:05:00 | 00:02:00 |
| | 13 | 00:02:00 | 00:03:00 | 00:01:00 | 00:02:00 | 00:03:00 |
| | 35 | - | 00:01:00 | 00:01:00 | - | 00:03:00 |
| Average | | 00:04:15 | 00:01:36 | 00:01:24 | 00:02:45 | 00:02:24 |

Table 6.4: Time used for informatics without documentation, to perform tasks on the **Adbudget** spreadsheet.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 33 | 00:06:00 | 00:02:00 | 00:01:00 | 00:05:00 | - |
| | 34 | - | 00:01:00 | 00:01:00 | 00:03:00 | 00:02:00 |
| Average | | 00:06:00 | 00:01:30 | 00:01:00 | 00:04:00 | 00:02:00 |

Table 6.5: Time used for **non-informatics with** documentation, to perform tasks on the **EnronGAS** spreadsheet.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 1 | 00:02:00 | 00:01:00 | 00:01:00 | 00:02:00 | 00:01:00 |
| | 3 | 00:03:00 | 00:01:00 | 00:01:00 | 00:02:00 | 00:01:00 |
| | 5 | 00:02:00 | 00:01:00 | 00:01:00 | 00:02:00 | 00:02:00 |
| | 25 | 00:02:00 | 00:01:00 | 00:01:00 | 00:01:00 | 00:02:00 |
| | 27 | 00:02:00 | 00:01:00 | 00:01:00 | 00:01:00 | 00:01:00 |
| Average | | 00:02:12 | 00:01:00 | 00:01:00 | 00:01:36 | 00:01:24 |

Table 6.6: Time used for **informatics** with documentation, to perform tasks on the **EnronGAS** spreadsheet.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 2 | 00:02:00 | 00:01:00 | 00:01:00 | 00:01:00 | 00:01:00 |
| | 26 | 00:02:00 | 00:01:00 | 00:01:00 | 00:04:00 | 00:02:00 |
| Average | | 00:02:00 | 00:01:00 | 00:01:00 | 00:02:30 | 00:01:30 |

Figure 6.2 and Figure 6.3 show the comparison of the time used for each participant between using our framework, and using the Excel, for Adbudget an EnronGas spreadsheet, respectively. Note that both figures represent the performances of **non-informatics** students to complete each task.

Table 6.7: Time used for **non-informatics without** documentation, to perform tasks on the **EnronGAS** spreadsheet.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 10 | 00:05:00 | 00:04:00 | 00:01:00 | 00:02:00 | 00:02:00 |
| | 11 | 00:02:00 | 00:03:00 | 00:01:00 | 00:01:00 | 00:02:00 |
| | 12 | 00:02:00 | 00:12:00 | 00:01:00 | 00:01:00 | 00:01:00 |
| | 13 | 00:07:00 | - | 00:01:00 | 00:01:00 | 00:02:00 |
| | 35 | - | - | 00:02:00 | 00:01:00 | - |
| Average | | 00:04:00 | 00:06:20 | 00:01:12 | 00:01:12 | 00:01:45 |

Table 6.8: Time used for **informatics without** documentation, to perform tasks on the **EnronGAS** spreadsheet.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 33 | 00:08:00 | 00:01:00 | 00:01:00 | 00:01:00 | 00:02:00 |
| | 34 | 00:14:00 | - | 00:01:00 | 00:01:00 | 00:04:00 |
| Average | | 00:11:00 | 00:01:00 | 00:01:00 | 00:01:00 | 00:03:00 |

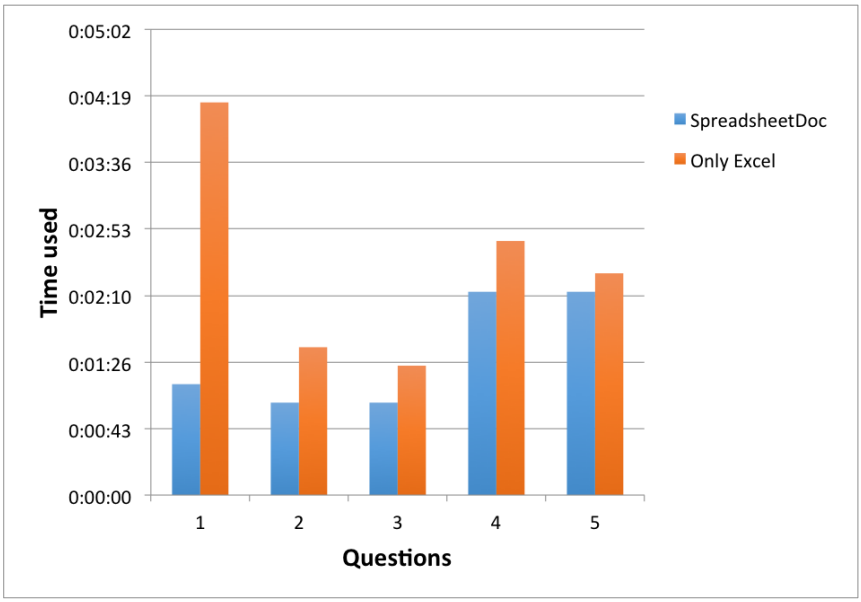



Figure 6.2: Time used to perform the tasks, for **non-informatics**, between SpreadsheetDoc and only Excel in **Adbugdet** spreadsheet.

Figure 6.4 and Figure 6.5 show the comparison of the time used for each participant between using our framework, and using the Excel, for Adbudget an EnronGas spreadsheet, respectively. Note that both figures represent the performances of **informatics** students to complete each task.

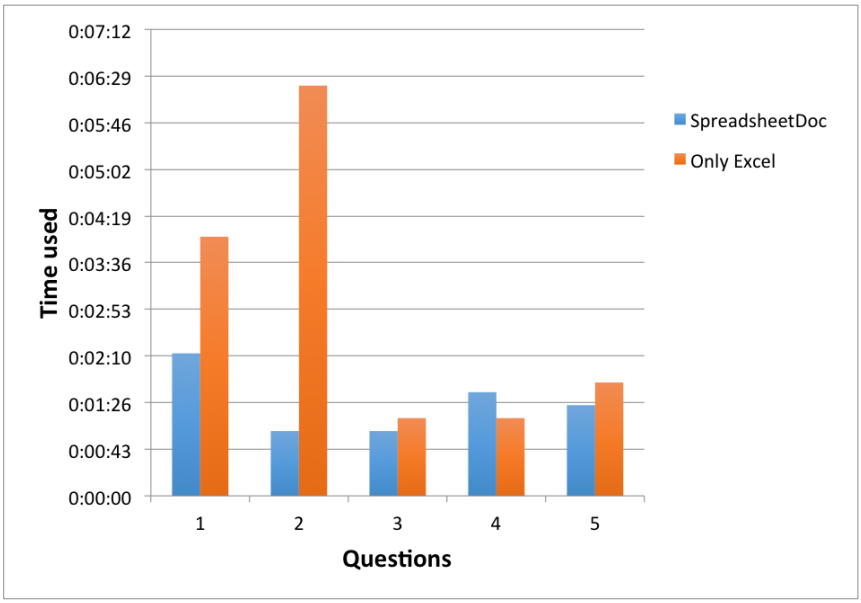



Figure 6.3: Time used to perform the tasks, for **non-informatics**, between SpreadsheetDoc and only Excel in **EnronGAS** spreadsheet.

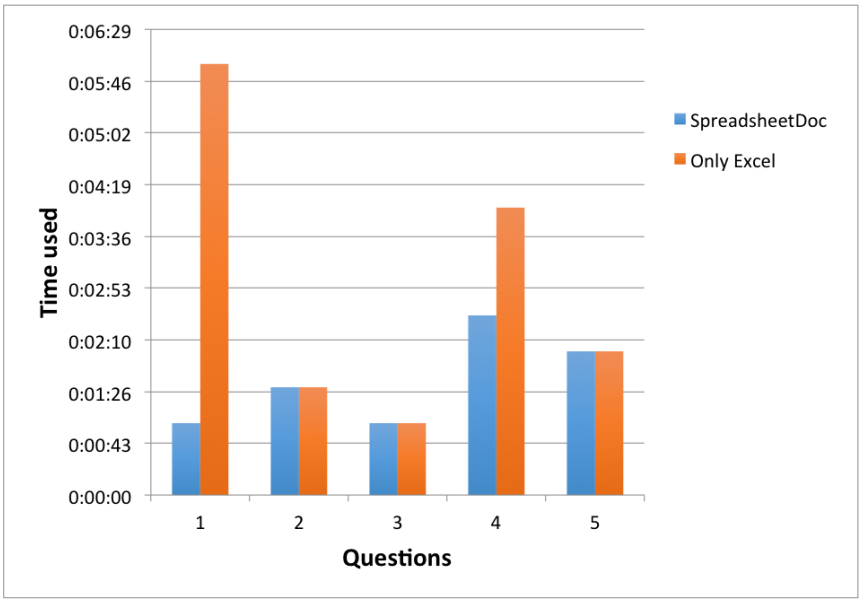



Figure 6.4: Time used to perform the tasks, for **informatics**, between SpreadsheetDoc and only Excel in **Adbudget** spreadsheet.

**Correctness grade (effectiveness):** To evaluate the correctness of the spreadsheets produced during the study, correctness grades are used. Each of the five tasks requires a set of spreadsheet operations to be correctly performed. Such operations included: change an input value and consult its result, identify the parameters (inputs) of a formula, and perform a modification in a formula. The correctness grade are evaluated using three values: 1, 0.5 and 0; where 1 is correct, 0.5 is half correct (e.g. insert an input correctly but consult the wrong cell) and 0 is incorrect.

Table 6.9 shows the correctness grade (effectiveness) of each non-informatics subject to

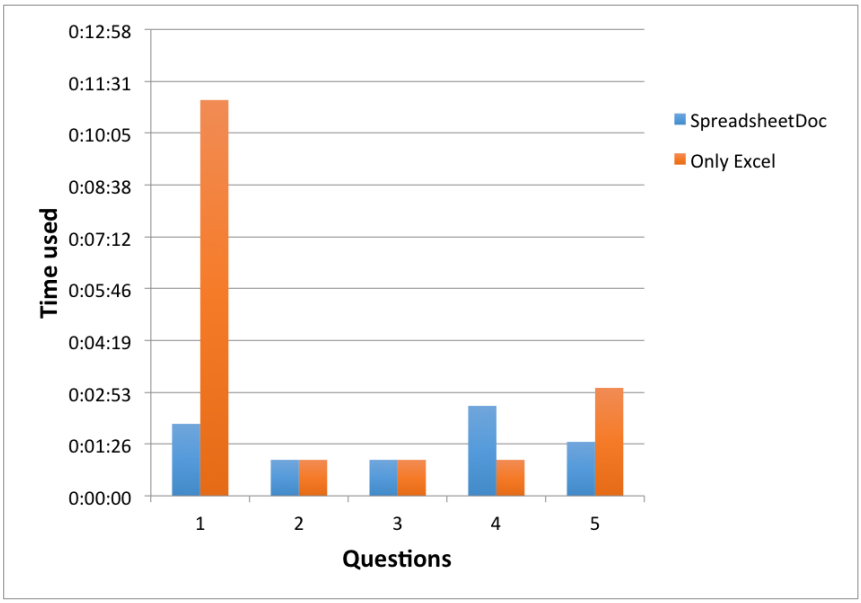



Figure 6.5: Time used to perform the tasks, for **informatics**, between SpreadsheetDoc and only Excel in **EnronGAS** spreadsheet.

perform their tasks on Adbudget spreadsheet. Table 6.10 shows the results using the SpreadsheetDoc while 6.11 shows the results using only the Excel. The first four rows for the question four have a hyphen (-) because they were cancelled due to an error on the spreadsheet. It was corrected for the next session that is why the last row has result. To the other questions, when it has a - it is because the subject did not answered.

Table 6.9: Correctness grade for **non-informatics** on the **Adbudget** spreadsheet.

Table 6.10: **With** documentation.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 1 | 1 | 1 | 1 | - | 1 |
| | 3 | 1 | 1 | 1 | - | 1 |
| | 5 | 1 | 1 | 1 | - | 1 |
| | 25 | 1 | 1 | 1 | - | 1 |
| | 27 | 1 | 1 | 1 | 1 | 1 |

Table 6.11: **Without** documentation.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 10 | 0 | 0.5 | 1 | - | 1 |
| | 11 | 0 | 0.5 | 1 | - | 1 |
| | 12 | 0 | 0.5 | 1 | - | 1 |
| | 13 | 0 | 0.5 | 1 | - | 1 |
| | 35 | - | 0.5 | 1 | - | 1 |

Table 6.12 shows the correctness grade (effectiveness) of each informatics subject to perform their tasks on Adbudget spreadsheet. Table 6.13 shows the results using the SpreadsheetDoc while 6.14 shows the results using only the Excel. When it has a hyphen (-) it is because the subject did not answered.

Table 6.15 shows the correctness grade (effectiveness) of each non-informatics subject to perform their tasks on EnronGAS spreadsheet. Table 6.16 shows the results using the SpreadsheetDoc while 6.17 shows the results using only the Excel. The first four rows for the question four have a hyphen (-) because they were cancelled due to an error on the spreadsheet. To the other questions and for the last row, when it has a - it is because the subject did not answered.

Table 6.12: Correctness grade for **informatics** on the **Adbudget** spreadsheet.

Table 6.13: **With** documentation.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 2 | 1 | 1 | 1 | 1 | 1 |
| | 26 | 1 | 1 | 0 | 1 | 1 |

Table 6.14: **Without** documentation.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 33 | 0 | 0.5 | 1 | 1 | - |
| | 34 | - | 0 | 1 | 0 | 1 |

Table 6.15: Correctness grade for **non-informatics** on the **EnronGAS** spreadsheet.

Table 6.16: **With** documentation.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 1 | 1 | 1 | 1 | 1 | 1 |
| | 3 | 1 | 1 | 1 | 1 | 1 |
| | 5 | 1 | 1 | 1 | 1 | 1 |
| | 25 | 1 | 1 | 1 | 1 | 1 |
| | 27 | 1 | 1 | 1 | 1 | 1 |

Table 6.17: **Without** documentation.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 10 | 0 | 0.5 | 1 | - | 1 |
| | 11 | 0 | 0.5 | 1 | - | 1 |
| | 12 | 0 | 0.5 | 1 | - | 1 |
| | 13 | 0 | 0.5 | 1 | - | 1 |
| | 35 | - | 0.5 | 1 | - | 1 |

Table 6.18 shows the correctness grade (effectiveness) of each informatics subject to perform their tasks on EnronGAS spreadsheet. Table 6.19 shows the results using the SpreadsheetDoc while 6.20 shows the results using only the Excel. When it has a hyphen (-) it is because the subject did not answered.

Table 6.18: Correctness grade for **informatics** on the **EnronGAS** spreadsheet.

Table 6.19: **With** documentation.

| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 2 | 1 | 1 | 1 | 1 | 1 |
| | 26 | 1 | 1 | 1 | 0 | 1 |

Table 6.20: **Without** documentation.

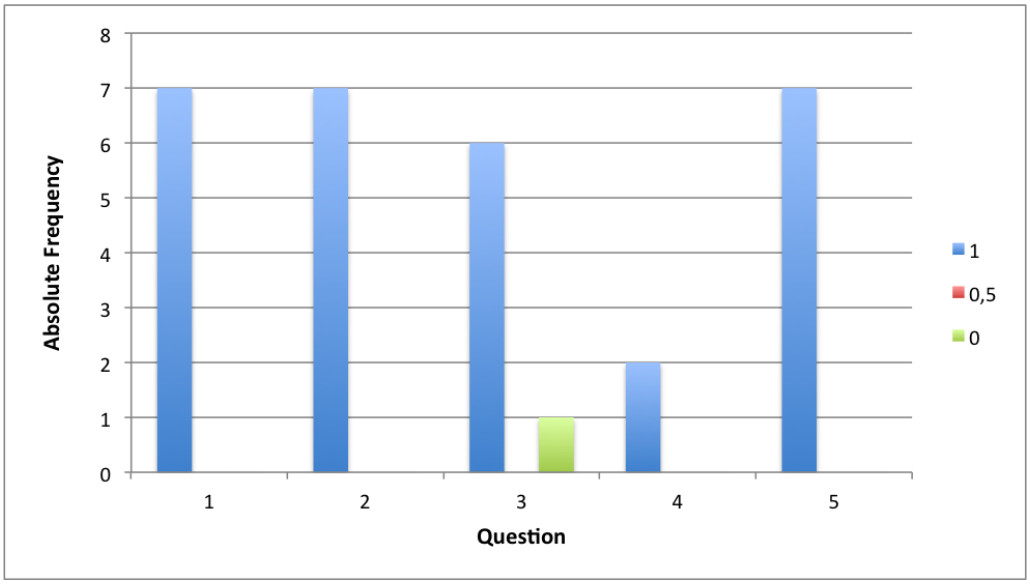
| | | Questions | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Participants | 33 | 1 | 1 | 1 | 1 | 1 |
| | 34 | 0.5 | - | 1 | 1 | 0 |

Figure 6.6 shows the absolute frequency (quantity of users' correctness grade, effectiveness) between subjects with documentation available (6.6(a)) and subjects that do not have documentation (6.6(b)) in Adbudget spreadsheet.

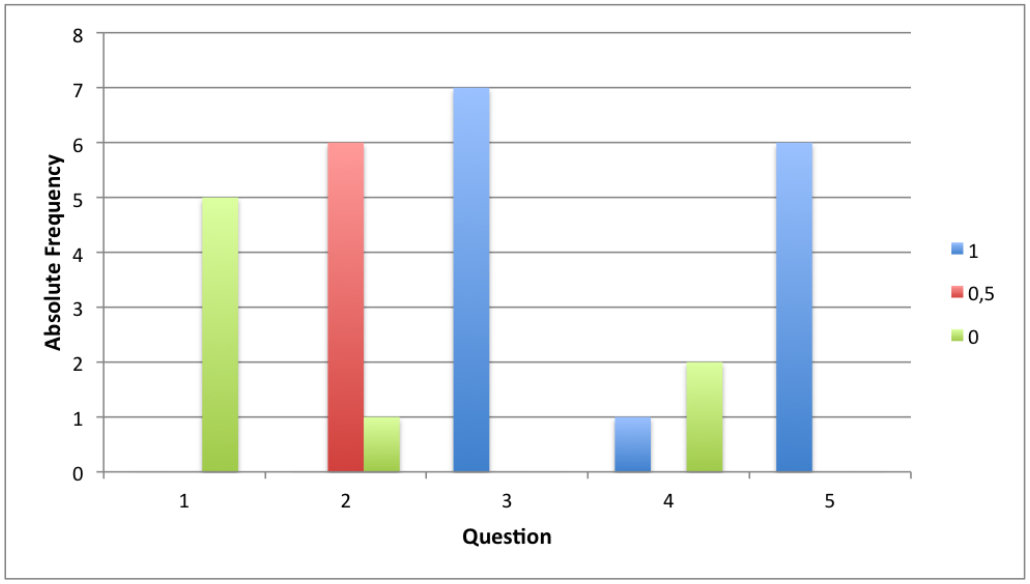
Figure 6.7 shows the absolute frequency (quantity of users' correctness grade) between subjects with documentation available (6.7(a)) and subjects that do not have documentation (6.7(b)) in EnronGAS spreadsheet.

### 6.3.2 Hypothesis Testing

The significance level used throughout the evaluation of all the tests is 0.05. The evaluation of the tests was performed using the SPSS software.




(a) Absolute frequency for **Adbudget** spreadsheet **with** documentation.

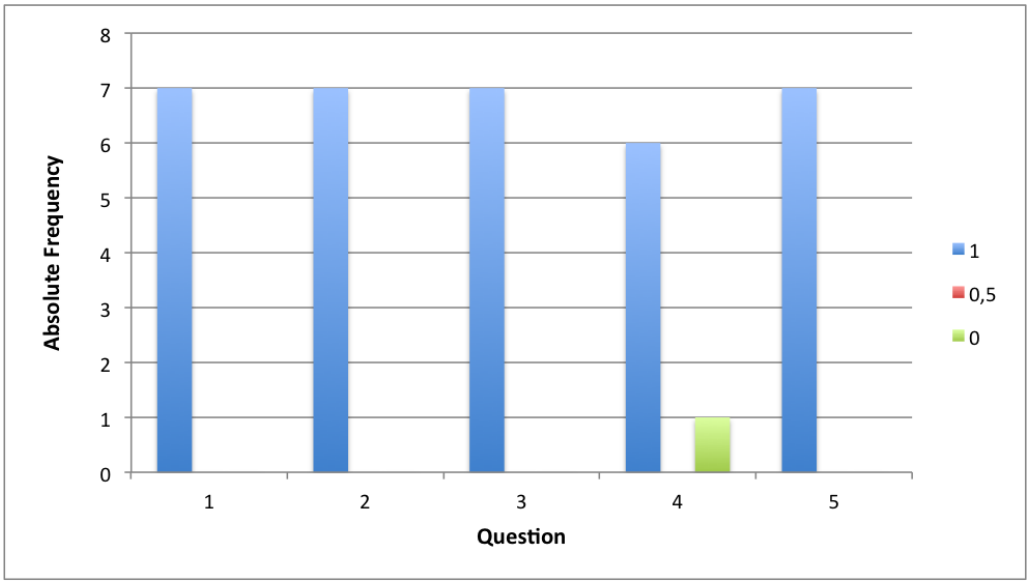



(b) Absolute frequency for **Adbudget** spreadsheet **without** documentation.

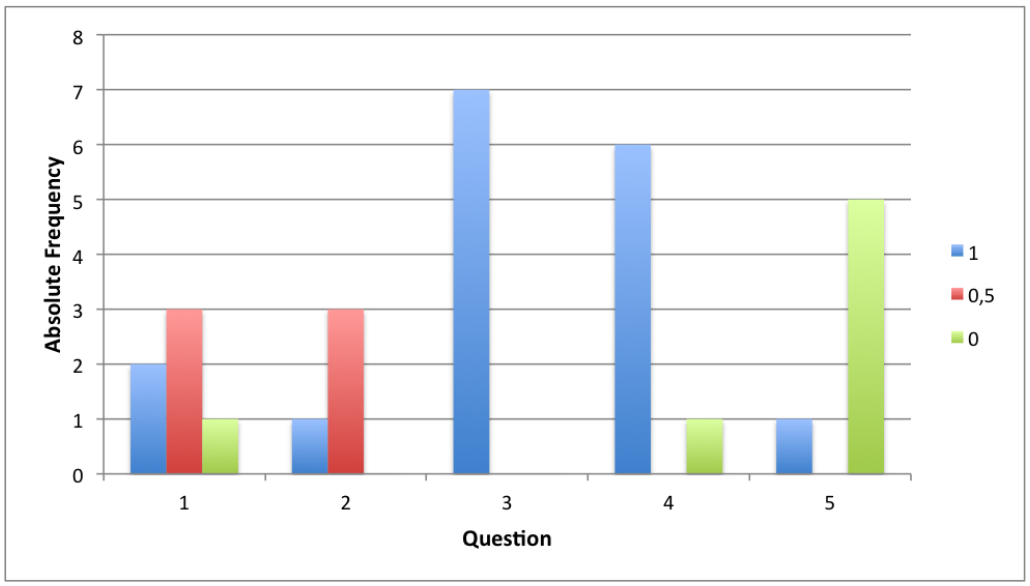
Figure 6.6: Absolute frequency of all subjects for **Adbudget** spreadsheet.

Since we have less participants than intended, we have group together some of the study results to perform the statistical analysis.

First, we have considered the answers to question $i$ (where $i \in \{1, 2, 3, 5\}$), both from spreadsheet EnronGAS and spreadsheet Adbudget, as answers to the same question, regarding the **time** participants took to answer them (but not for correctness). We did this as the questions are of the same kind and quite similar, being the only difference the spreadsheet they are written for. Moreover, this is possible because there is no statistical difference between the time participants took to answer the questions from both spreadsheets. Indeed, a Mann-Whitney test indicated that the times spent in question $i$ (where $i \in \{1, 2, 3, 5\}$) was not significantly different for the Adbudget spreadsheet ($Mdn_1$ = 00:02, $Mdn_2$ = 00:01, $Mdn_3$ = 00:01, $Mdn_5$ =




(a) Absolute frequency for **EnronGAS** spreadsheet **with** documentation.




(b) Absolute frequency for **EnronGAS** spreadsheet **without** documentation.

Figure 6.7: Absolute frequency of all subjects for **EnronGAS** spreadsheet.

00:02), and EnronGAS ($Mdn_1$ = 00:02, $Mdn_2$ = 00:01, $Mdn_3$ = 00:01, $Mdn_5$ = 00:02), $U_1 = 67$, $U_2 = 72.5$, $U_3 = 91$, $U_5 = 74.5$, $p_1 = .518$, $p_2 = .755$, $p_3 = .549$, $p_5 = .581$.[1] There was however a significantly difference for question 4 and thus their times cannot be used together.

Second, we have also grouped the results of both informatics and non-informatics. This is only possible as there is no statistical difference between the answers of informatics and others. Indeed, a Chi-square test of independence was calculated comparing the frequency of each kind of answer (correct, half-correct, and incorrect) for question $i$ (where $i \in \{1, 2, 3, 5\}$) for both spreadsheets between informatics and non-informatics. The test did not found a significant

[1] We follow the reporting format suggested in [2].

interaction ($p_1 = .936$, $p_2 = .300$, $p_3 = .107$, $p_5 = .557$).[2] This was calculated using answers from both spreadsheets together. Since that is not possible for question 4, we do not use such result here.

**Comparison of correctness** We have performed three tests (all suggested by SPSS) to verify the statistical significance of the correctness grade: Kendall's tau-b, Kendall's tau-c, and Gamma.

For the EnronGAS spreadsheet the correctness grade is greater when using SpreadsheetDoc for questions 1, 2, and 5 (p = .000 for the three questions and for the three tests).

For the Adbudget spreadsheet the correctness grade is greater when using SpreadsheetDoc for questions 1, 2, and 4 (p = .000 for the three tests for question 1, p = .048 for Kendall's tau-b test and p = .000 for the others for question 2, and p = .034 for the three tests for question 4).

For the others it is not possible to determine the statistical significance of the differences between the answers.

Table 6.21 resumes the statistical significance of the answers for each spreadsheet, as well as when considering both spreadsheets together. The hyphen mark is used to denote the impossibility of merging the results.

Table 6.21: Statistical relevance for **answers**.

| | Questions | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| Adbudget | ✓ | ✓ | ✗ | ✓ | ✓ |
| EnronGAS | ✓ | ✓ | ✗ | ✗ | ✓ |
| Together | - | ✓ | ✗ | ✗ | - |

**Comparison of times** In this case, to verify the statistical relevance, we have used the test Mann-Whitney.

For the EnronGAS spreadsheet the Mann-Whitney test indicated that the time spent to answer question $i$ (where $i \in \{1, 2\}$) was greater without the use of SpreadsheetDoc ($Mdn_1$ = 00:06, $Mdn_4$ = 00:03), than using SpreadsheetDoc ($Mdn_1$ = 00:02, $Mdn_4$ = 00:01), $U_1 = 8$, $U_4 = 3.5$, $p_1 = .034$, $p_4 = .012$.

For the Adbudget spreadsheet the Mann-Whitney test indicated that the time spent in question 1 without SpreadsheetDoc ($Mdn$ = 00:05) was greater than with SpreadsheetDoc ($Mdn$ = 00:02), $U = 3.5$, $p = .034$.

As we described it is statistically possible to merge both spreadsheet results. In this case, an extra significance arises for question 5. Indeed, for both spreadsheets together, the Mann-Whitney test indicated that the time spent in question 5 without SpreadsheetDoc ($Mdn$ = 00:05) was greater than with SpreadsheetDoc ($Mdn$ = 00:02), $U = 42.5$, $p = .02$. There is also significance for questions 1 and 2, but that already happened for each spreadsheet individually.

For the remaining cases it is not possible to determine the statistical significance of the differences between the times participants took to answer the questions.

[2] Again, we follow the reporting format suggested in [2].

Table 6.22 resumes the statistical significance of times spent in the answers for each spreadsheet, as well as when considering both spreadsheets together. Again, the hyphen mark is used to denote the impossibility of merging the results.

Table 6.22: Statistical relevance for **times**.

| | Questions | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| Adbudget | ✓ | ✗ | ✗ | ✗ | ✓ |
| EnronGAS | ✓ | ✓ | ✗ | ✗ | ✓ |
| Together | ✓ | ✓ | ✗ | - | ✓ |

## 6.4 Interpretation

The results from the analysis suggest that SpreadsheetDoc can improve users' performance doing their tasks. However, this was not the case for question three. We believe this is because the answer to this question is quite easy. Users just have to consult the formula arguments and in that way, our framework does not provide any extra help. In question four of EnronGAS we have had interesting results. Users without documentation were more efficients that the other, but in the other hand their effectiveness was worst when compared to users with documentation. So, in this question we can conclude that SpreadsheetDoc helps users to perform the tasks correctly in spite of taken more time.

Comparing the efficiency and effectiveness registered on the tables it is possible to conclude that our framework was useful mainly for questions one and two.

Moreover, from the post-questionnaire we can conclude that subjects felt more confident in the results using our framework when compared to the ones who used only Excel. This indicates that our framework is useful and some given suggestions were welcome.

### 6.4.1 Threats to validity

The goal of the study is to demonstrate that is better use our framework than only Excel. Therefore, validity threats for this study were analysed and divided in four categories as defined in [4]: conclusion validity, internal validity, construct validity and external validity.

**Conclusion validity:** The main concern is the low statistical power due to the low number of participants. To overcome this issue we have decided to group the same question of each spreadsheet and which could improve our statistical power. It is important to note that this only could be possible due to a statistical test which allowed it.

**Internal validity:** In order to minimize the effects on the independent variables that would reflect on the causality, several actions were taken. First, this study was executed several times (different sessions), where the subjects of one session work with our framework and others did not. Second, within each session, some participants have started with the Adbudget spreadsheet and others with the EnronGAS spreadsheet to minimize learning effects. Third, the time to

perform the study was reduced as much as possible so that the subjects could remain focused during all the study (fifteen minutes for each spreadsheet). Fourth, all the subjects performed the same tasks, so issues from having different groups with distinct treatments do not arise. This specifications intend to obtain more control and reduce internal validity threats.

**Construct validity:** For this validity we have used two spreadsheets. Before starting performing the tasks, we have informed the subjects that they were not under evaluation guaranteeing that they were not affected by this study.

Some subjects have done the study using documentation and others without it and the tasks we have asked users to perform are common in spreadsheets, such as insert values, consult values, and change formulas. So, we believe the study construction allows to evaluate the use of SpreadsheetDoc.

**External validity:** This validity is related to the strength to generalize the results of this study to industrial practice. Due to this, we have selected two spreadsheets from the real-world: one from a company and another from a book on the design of spreadsheets. Although the spreadsheets are real-world spreadsheets, the environment is not. Nevertheless, the participants represent a wide range of spreadsheet users, and thus, we believe that results are generalizable.

### 6.4.2 Inferences

Since this study was performed in a very specific environment, we cannot generalize it to every case. Nevertheless, the environment used to perform this study was as similar as possible to a real one, in which end users are normally non-professionals and in which spreadsheets are already developed with a specific purpose. Therefore, the used spreadsheets were based on real cases, and the majority of the students which preformed the study were from a non-informatics areas.

Our framework was developed mainly for end users, so it could be useful if applied on a professional industry.

## 6.5 Discussion

The empirical study we have conducted reveals very promising results for our framework, SpreadsheetDoc. Despite that participants had a shorter period of time to learn on a more efficient way of how to work with our framework, they have accomplished their tasks faster when compared to the participants that have not used our framework.

As we have envisioned, it was apparently easy to use our framework and useful. Most participants wrote on post-questionnaire: “This tool help our understanding of the spreadsheet” or “Very useful framework”.

We could conclude that our framework improves users’ efficiency and effectiveness. So, our framework have had and improvement around of 50% of the times. As usual, there was exceptions. We can also conclude that computer science using our framework or not have had similar results, but our framework still has had better results.

Chapter 7

# State of the Art

In this chapter we present the few approaches currently existent to document spreadsheets. In Section 7.1 we describe a system for developers to document spreadsheets. In Section 7.2 we discuss how users document a spreadsheet, and in Section 7.3 we discuss what Excel allows.

## 7.1 Programming Documentation

In [22] the authors present a system to format spreadsheet documentation. This system uses an external editor to document spreadsheets and macros to format such text. This can be compared to literate programming, where documentation and code are kept together in the same file. The objective is to easily write and update documentation for spreadsheets. However, the user must "program" the documentation itself, for instance, writing the following code to create a variable to document a formula: `@MACRO(xfor(v)=[@P(@V(f_@X(v)))])`. These variables can then be assigned to parts of the spreadsheet and used in the textual documentation. However, this seems difficult to learn, specially for end users. Our approach on the other hand shows contextualized wizards with the necessary text inputs the documentation writer must fill in, making it more convenient for end users to document their spreadsheets.

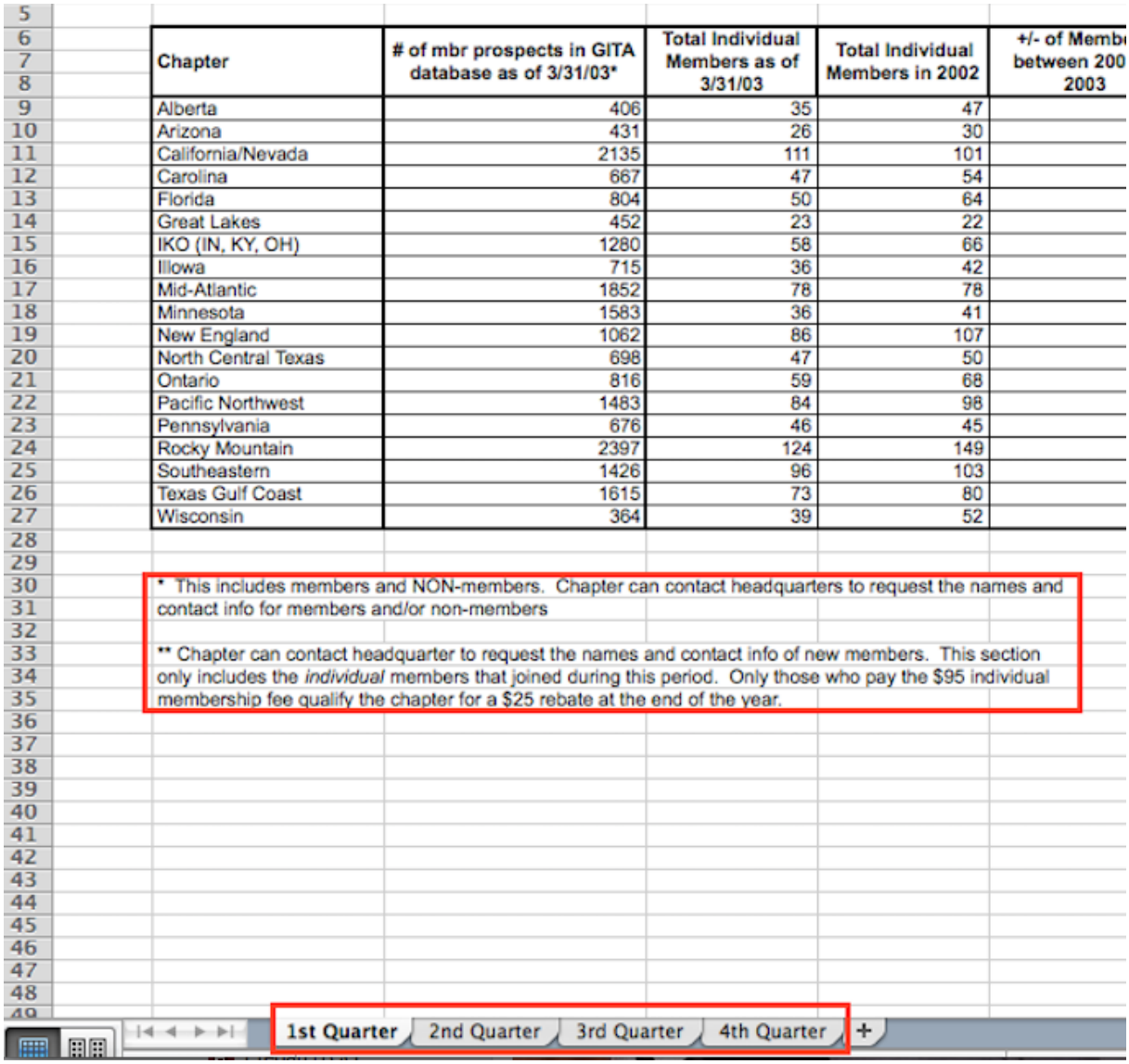
## 7.2 Ad-hoc Documentation

We searched on different spreadsheets with the purpose of finding distinct types of documentation. Indeed, we found what we were searching for, but such documentation is too rudimentary, that is, it is not structured.

Therefore, we are going to give some examples of what users do and explain the examples.

One way users document spreadsheets is by commenting below the presented content. In Figure 7.1, users document below the table. They explain the meaning of each asterisk. This is a type of documentation very common in spreadsheets. In this example, the user divided the documentation by worksheet. In fact, there are four worksheets and each one has its own

documentation. With our approach users can select a range of cells and document it, for instance, the table in this example is in range the B6:F27.



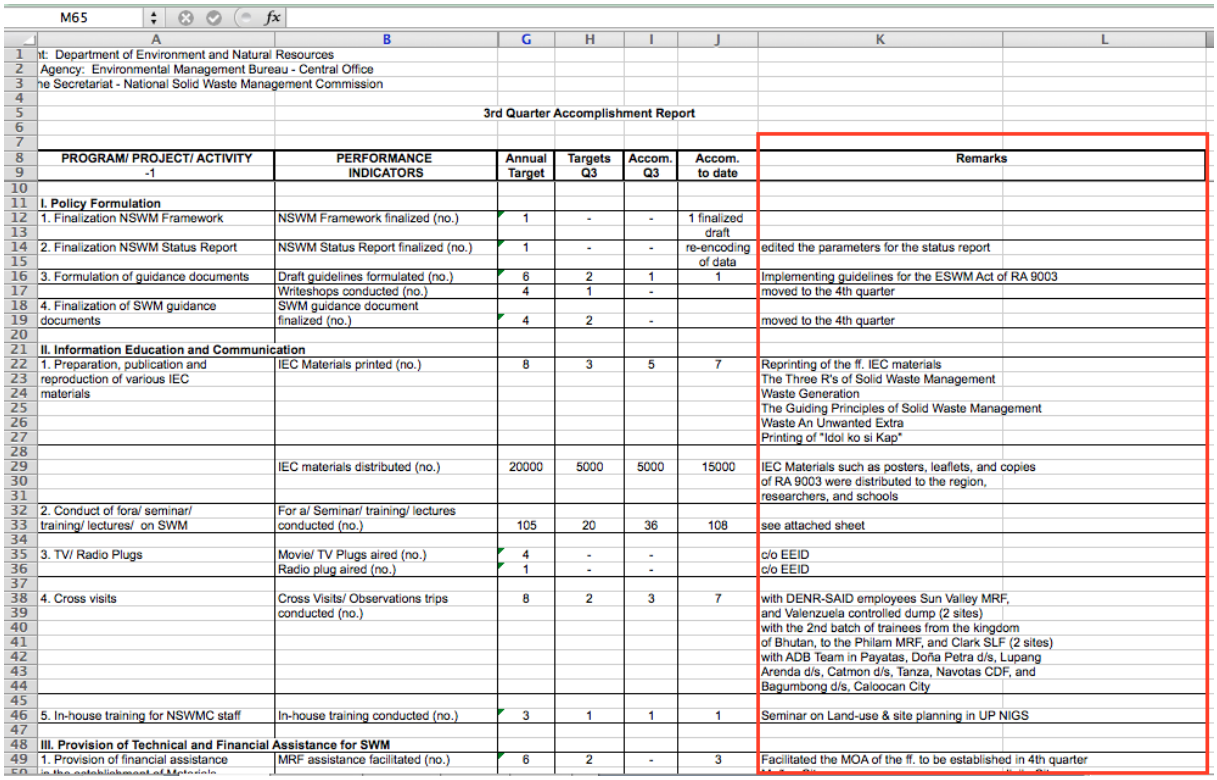
| Chapter | # of mbr prospects in GITA database as of 3/31/03* | Total Individual Members as of 3/31/03 | Total Individual Members in 2002 | +/- of Memb between 200 2003 |
|---|---|---|---|---|
| Alberta | 406 | 35 | 47 | |
| Arizona | 431 | 26 | 30 | |
| California/Nevada | 2135 | 111 | 101 | |
| Carolina | 667 | 47 | 54 | |
| Florida | 804 | 50 | 64 | |
| Great Lakes | 452 | 23 | 22 | |
| IKO (IN, KY, OH) | 1280 | 58 | 66 | |
| Illowa | 715 | 36 | 42 | |
| Mid-Atlantic | 1852 | 78 | 78 | |
| Minnesota | 1583 | 36 | 41 | |
| New England | 1062 | 86 | 107 | |
| North Central Texas | 698 | 47 | 50 | |
| Ontario | 816 | 59 | 68 | |
| Pacific Northwest | 1483 | 84 | 98 | |
| Pennsylvania | 676 | 46 | 45 | |
| Rocky Mountain | 2397 | 124 | 149 | |
| Southeastern | 1426 | 96 | 103 | |
| Texas Gulf Coast | 1615 | 73 | 80 | |
| Wisconsin | 364 | 39 | 52 | |



Figure 7.1: A spreadsheet taken from EUSES, with path database/processed/03Quarterly ChapterA840A.xls.

Another way is to comment besides the content. It may have a header, for instance, "remark" or other synonym. In Figure 7.2, users create a header named "remarks". So, they are adding to the spreadsheet more information because they have no other way to document it. Each row can be commented as shown. Some cells have comments more extensive than other. It describes the row's behaviour. Our approach allows users to document each row they want.

In the first two examples, users are inserting extra cells in the spreadsheet, which are not part of the program, making it more complex.



| | A | B | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|
| 1 | nt: Department of Environment and Natural Resources | | | | | | | |
| 2 | Agency: Environmental Management Bureau - Central Office | | | | | | | |
| 3 | he Secretariat - National Solid Waste Management Commission | | | | | | | |
| 4 | | | | | | | | |
| 5 | | 3rd Quarter Accomplishment Report | | | | | | |
| 6 | | | | | | | | |
| 7 | | | | | | | | |
| 8 | PROGRAM/ PROJECT/ ACTIVITY -1 | PERFORMANCE INDICATORS | Annual Target | Targets Q3 | Accom. Q3 | Accom. to date | Remarks | |
| 10 | | | | | | | | |
| 11 | I. Policy Formulation | | | | | | | |
| 12 | 1. Finalization NSWM Framework | NSWM Framework finalized (no.) | 1 | - | - | 1 finalized draft | | |
| 14 | 2. Finalization NSWM Status Report | NSWM Status Report finalized (no.) | 1 | - | - | re-encoding of data | edited the parameters for the status report | |
| 16 | 3. Formulation of guidance documents | Draft guidelines formulated (no.) | 6 | 2 | 1 | 1 | Implementing guidelines for the ESWM Act of RA 9003 | |
| 17 | | Writeshops conducted (no.) | 4 | 1 | - | | moved to the 4th quarter | |
| 18 | 4. Finalization of SWM guidance | SWM guidance document | | | | | | |
| 19 | documents | finalized (no.) | 4 | 2 | - | | moved to the 4th quarter | |
| 20 | | | | | | | | |
| 21 | II. Information Education and Communication | | | | | | | |
| 22 | 1. Preparation, publication and | IEC Materials printed (no.) | 8 | 3 | 5 | 7 | Reprinting of the ff. IEC materials | |
| 23 | reproduction of various IEC | | | | | | The Three R's of Solid Waste Management | |
| 24 | materials | | | | | | Waste Generation | |
| 25 | | | | | | | The Guiding Principles of Solid Waste Management | |
| 26 | | | | | | | Waste An Unwanted Extra | |
| 27 | | | | | | | Printing of "Idol ko si Kap" | |
| 28 | | | | | | | | |
| 29 | | IEC materials distributed (no.) | 20000 | 5000 | 5000 | 15000 | IEC Materials such as posters, leaflets, and copies | |
| 30 | | | | | | | of RA 9003 were distributed to the region, | |
| 31 | | | | | | | researchers, and schools | |
| 32 | 2. Conduct of fora/ seminar/ | For a/ Seminar/ training/ lectures | | | | | | |
| 33 | training/ lectures/ on SWM | conducted (no.) | 105 | 20 | 36 | 108 | see attached sheet | |
| 34 | | | | | | | | |
| 35 | 3. TV/ Radio Plugs | Movie/ TV Plugs aired (no.) | 4 | - | - | | c/o EEID | |
| 36 | | Radio plug aired (no.) | 1 | - | - | | c/o EEID | |
| 37 | | | | | | | | |
| 38 | 4. Cross visits | Cross Visits/ Observations trips | 8 | 2 | 3 | 7 | with DENR-SAID employees Sun Valley MRF, | |
| 39 | | conducted (no.) | | | | | and Valenzuela controlled dump (2 sites) | |
| 40 | | | | | | | with the 2nd batch of trainees from the kingdom | |
| 41 | | | | | | | of Bhutan, to the Philam MRF, and Clark SLF (2 sites) | |
| 42 | | | | | | | with ADB Team in Payatas, Doña Petra d/s, Lupang | |
| 43 | | | | | | | Arenda d/s, Catmon d/s, Tanza, Navotas CDF, and | |
| 44 | | | | | | | Bagumbong d/s, Caloocan City | |
| 45 | | | | | | | | |
| 46 | 5. In-house training for NSWMC staff | In-house training conducted (no.) | 3 | 1 | 1 | 1 | Seminar on Land-use & site planning in UP NIGS | |
| 47 | | | | | | | | |
| 48 | III. Provision of Technical and Financial Assistance for SWM | | | | | | | |
| 49 | 1. Provision of financial assistance | MRF assistance facilitated (no.) | 6 | 2 | - | 3 | Facilitated the MOA of the ff. to be established in 4th quarter | |

Figure 7.2: A spreadsheet taken from EUSES, with path database/processed/3rd-20quarter.xls.

## 7.3 Excel Documentation

Excel is the most used program to develop spreadsheets, and it has some ways of doing documentation. In Figure 7.3 we show the properties available to document a spreadsheet such as author, title, spreadsheet subject, etc. In fact, users can add some comments too, but not much more.

Another example is the insertion of a comment on a selected cell; the comment is attached to the cell. In Figure 7.4 users insert a comment on the selected cell. It is a feature available on Excel. When the user passes the mouse over the cell, a pop up is opened and it is possible to read the comment. It is not very efficient to use because when users are working on a spreadsheet with big amount of data, it becomes too confusing due to the huge amount of documentation non structured.

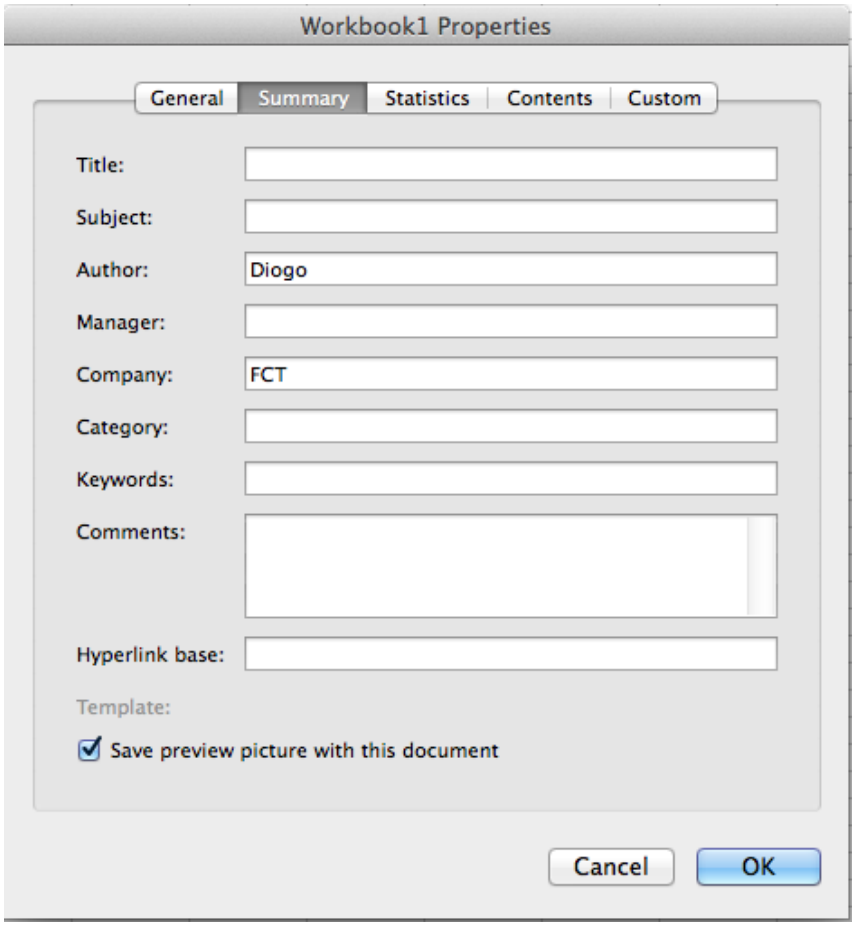



Figure 7.3: Excel properties of a spreadsheet.

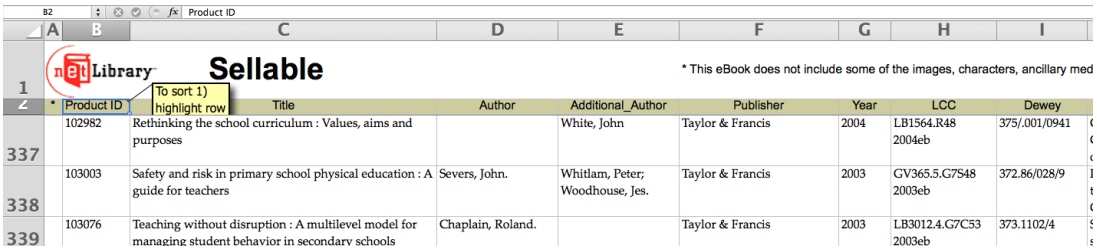



Figure 7.4: A spreadsheet taken from EUSES, with path database/processed/Coll2first-load.xls.

# Chapter 8

# Conclusions

This is the last chapter we present some conclusion remarks (Section 8.1) and future work (Section 8.2).

## 8.1 Concluding Observations

Spreadsheets are the most used programming environment in the world. However, they lack many of the features modern programming environments offer. In particular, there is no structured way of documenting spreadsheet programs. Indeed, there is strong evidence that users waste too much time trying to understand spreadsheets, especially when they are using one that was not created by them. Hence, they search within spreadsheets trying to find some kind of documentation, ask for help to colleagues, or end up by quitting.

To alleviate this scenario, we have developed a framework, SpreadsheetDoc, to permit the writing and reading documentation of spreadsheets, which may be very useful when used correctly. We have implemented our approach as an add-in for Excel that allows users to write and read documentation through a set of wizards to facilitate its usage.

Indeed, the empirical validation we performed gives strength to our framework because in most cases we can indeed improve users' performance. Moreover, the opinions of participants are mostly positive.

Nevertheless, there are still several issues we would like to address in future work, which we discuss in the next section.

## 8.2 Future Work

In our approach we did not work on documenting the VBA scripts that are part of some spreadsheets. Although this may seem important, in the Enron's corpus only 47 spreadsheets (out of more than 15.000) used VBA scripts [9]. Also in the EUSES corpus only 126 (out of 4.498) used VBA [6]. Nevertheless, we will also address this in future work. Since in this case we are

probably addressing more advanced users, we intend to follow an approach similar to JavaDoc, where the programmer annotates the different parts of the source code, and from which it is possible to generate a comprehensive web page. This will extend the web page generated by our tool.

We foresee several improvements for SpreadsheetDoc. To make this framework even more interesting we should link the input of a formula cell to the wizard to documented it.

Another quite interesting and promising direction is the automation or inference of documentation using the system presented in [1]. This system is more robust than the system used by us, that is, it covered all the cases of inference of headers and units.

Moreover, we would also like to integrate some heuristics to automatically describe existing formulas, in a similar way as described in [20]. For instance, for our running example, the system could automatically infer the following description for cell I10: *Cell I10 calculates Win? No, that is, the maximum between Profit new ship and Profit tug/barge.* This can easily be inferred from the formula, its inputs, and corresponding labels.

# Bibliography


[1] R. Abraham and M. Erwig. "Header and Unit Inference for Spreadsheets Through Spatial Analyses". In: *Proceedings of the 2004 IEEE Symposium on Visual Languages - Human Centric Computing*. VLHCC '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 165–172. ISBN: 0-7803-8696-5. DOI: 10.1109/VLHCC.2004.29. URL: http://dx.doi.org/10.1109/VLHCC.2004.29.

[2] L. Burnham. *Reporting Statistics in Psychology*. Online. accessed 29-March-2016. URL: http://evc-cit.info/psych018/Reporting_Statistics.pdf.

[3] D. Canteiro and J. Cunha. "SpreadsheetDoc: An Excel Add-in for Documenting Spreadsheets". In: *Proceedings of the 6th National Symposium of Informatics (INForum'15)*. Covilhã, Portugal, 2015, pp. 1–16.

[4] T. D. Cook, D. T. Campbell, and A. Day. *Quasi-experimentation: Design & analysis issues for field settings*. Vol. 351. Houghton Mifflin Boston, 1979.

[5] J. Cunha, J. P. Fernandes, J. Mendes, and J. Saraiva. "Embedding, Evolution, and Validation of Spreadsheet Models in Spreadsheet Systems". In: *IEEE Transactions on Software Engineering* 43.3 (2014), pp. 241–263. ISSN: 0098-5589. DOI: 10.1109/TSE.2014.2361141. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6915751.

[6] M. Fisher and G. Rothermel. "The EUSES Spreadsheet Corpus: A Shared Resource for Supporting Experimentation with Spreadsheet Dependability Mechanisms". In: *Proceedings of the First Workshop on End-user Software Engineering*. WEUSE I. St. Louis, Missouri: ACM, 2005, pp. 1–5. ISBN: 1-59593-131-7. DOI: 10.1145/1082983.1083242. URL: http://doi.acm.org/10.1145/1082983.1083242.

[7] K. A. de Graaf, P. Liang, A. Tang, and H. van Vliet. "Supporting Architecture Documentation: A Comparison of Two Ontologies for Knowledge Retrieval". In: *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*. EASE '15. Nanjing, China: ACM, 2015, 3:1–3:10. ISBN: 978-1-4503-3350-4. DOI: 10.1145/2745802.2745804. URL: http://doi.acm.org/10.1145/2745802.2745804.

[8] F. Hermans. "Gathering Domain Knowledge from Spreadsheets". In: *Proceedings of the Doctoral Symposium for ESEC/FSE on Doctoral Symposium*. ESEC/FSE Doctoral Symposium '09. Amsterdam, The Netherlands: ACM, 2009, pp. 37–38. ISBN: 978-1-60558-731-8. DOI: 10.1145/1595782.1595798. URL: http://doi.acm.org/10.1145/1595782.1595798.

[9] F. Hermans and E. Murphy-Hill. "Enron's Spreadsheets and Related Emails: A Dataset and Analysis". In: *Proceedings of the 37th International Conference on Software Engineering - Volume 2*. ICSE '15. Florence, Italy: IEEE Press, 2015, pp. 7–16. URL: http://dl.acm.org/citation.cfm?id=2819009.2819013.

[10] F. Hermans, M. Pinzger, and A. van Deursen. "Breviz: Visualizing Spreadsheets using Dataflow Diagrams". In: *CoRR* abs/1111.6895 (2011). URL: http://arxiv.org/abs/1111.6895.

[11] F. Hermans, M. Pinzger, and A. van Deursen. "Supporting Professional Spreadsheet Users by Generating Leveled Dataflow Diagrams". In: *Proceedings of the 33rd International Conference on Software Engineering*. ICSE '11. Waikiki-Honolulu, HI, USA: ACM, 2011, pp. 451–460. ISBN: 978-1-4503-0445-0. DOI: 10.1145/1985793.1985855. URL: http://doi.acm.org/10.1145/1985793.1985855.

[12] B. Kankuzi and J. Sajaniemi. "A domain terms visualization tool for spreadsheets". In: *Visual Languages and Human-Centric Computing (VL/HCC), 2014 IEEE Symposium on*. 2014, pp. 209–210. DOI: 10.1109/VLHCC.2014.6883059.

[13] B. Klimt and Y. Yang. "Introducing the Enron Corpus". In: *CEAS 2004 - First Conference on Email and Anti-Spam, July 30-31, 2004, Mountain View, California, USA*. 2004. URL: http://www.ceas.cc/papers-2004/168.pdf.

[14] D. Kramer. "API Documentation from Source Code Comments: A Case Study of Javadoc". In: *Proceedings of the 17th Annual International Conference on Computer Documentation*. SIGDOC '99. New Orleans, Louisiana, USA: ACM, 1999, pp. 147–153. ISBN: 1-58113-072-4. DOI: 10.1145/318372.318577. URL: http://doi.acm.org/10.1145/318372.318577.

[15] J. Lawrance, R. Abraham, M. M. Burnett, and M. Erwig. "Sharing reasoning about faults in spreadsheets: An empirical study". In: *2006 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2006), 4-8 September 2006, Brighton, UK*. IEEE Computer Society, 2006, pp. 35–42. ISBN: 0-7695-2586-5. DOI: 10.1109/VLHCC.2006.43. URL: http://doi.ieeecomputersociety.org/10.1109/VLHCC.2006.43.

[16] B. van Loggem. "Software Documentation: A Standard for the 21st Century". In: *Proceedings of the International Conference on Information Systems and Design of Communication*. ISDOC '14. Lisbon, Portugal: ACM, 2014, pp. 149–154. ISBN: 978-1-4503-2713-8. DOI: 10.1145/2618168.2618192. URL: http://doi.acm.org/10.1145/2618168.2618192.

[17] S. G. Powell and K. R. Baker. *The Art of Modeling with Spreadsheets*. New York, NY, USA: John Wiley & Sons, Inc., 2003. ISBN: 0471209376.

[18] K. Rajalingham, D. R. Chadwick, and B. Knight. "Classification of Spreadsheet Errors". In: *CoRR* abs/0805.4224 (2008). URL: http://arxiv.org/abs/0805.4224.

[19] J. G. Raymond. "Audience Identification for End User Documentation". In: *Proceedings of the June 7-10, 1982, National Computer Conference*. AFIPS '82. Houston, Texas: ACM, 1982, pp. 281–285. ISBN: 0-88283-035-X. DOI: 10.1145/1500774.1500808. URL: http://doi.acm.org/10.1145/1500774.1500808.

[20] S. Roy. "Business Rule Mining from Spreadsheets". In: *Proceedings of the Second Workshop on Software Engineering Methods in Spreadsheets co-located with the 37th International Conference on Software Engineering (ICSE 2015)*. Ed. by F. Hermans, R. F. Paige, and P. Sestof. Vol. 1355. SEMS '15. CEUR, 2015, pp. 5–6.

[21] J. E. Scott. "Technology Acceptance and ERP Documentation Usability". In: *Commun. ACM* 51.11 (Nov. 2008), pp. 121–124. ISSN: 0001-0782. DOI: 10.1145/1400214.1400239. URL: http://doi.acm.org/10.1145/1400214.1400239.

[22] R. M. Snyder. "A System for Automating the Update of Spreadsheet Documentation". In: *J. Comput. Sci. Coll.* 23.2 (Dec. 2007), pp. 163–169. ISSN: 1937-4771. URL: http://dl.acm.org/citation.cfm?id=1292428.1292456.

[23] TIOBE. *TIOBE Index | Tiobe - The Software Quality Company*. Online. accessed 12-March-2016. 2016. URL: http://www.tiobe.com/tiobe_index.

Appendix

# XSD Schema

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Spreadsheet">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Worksheet" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:choice maxOccurs="unbounded" minOccurs="0">
              <xs:element name="Cell">
                <xs:complexType mixed="true">
                  <xs:sequence>
                    <xs:element name="input" maxOccurs="unbounded"
                    minOccurs="0">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension base="xs:string">
                            <xs:attribute type="xs:string"
                            name="cell" use="optional"/>
                            <xs:attribute type="xs:string"
                            name="description" use="optional"/>
                          </xs:extension>
                        </xs:simpleContent>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="output" minOccurs="0">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension base="xs:string">
                            <xs:attribute type="xs:string" name="description"
                            use="optional"/>
                          </xs:extension>
                        </xs:simpleContent>
                      </xs:complexType>
```

```
34                    </xs:element>
35                  </xs:sequence>
36                  <xs:attribute type="xs:string" name="name"
37                  use="optional"/>
38                  <xs:attribute type="xs:string" name="description"
39                  use="optional"/>
40                </xs:complexType>
41              </xs:element>
42              <xs:element name="Input">
43                <xs:complexType>
44                  <xs:simpleContent>
45                    <xs:extension base="xs:string">
46                      <xs:attribute type="xs:string" name="name"
47                      use="optional"/>
48                      <xs:attribute type="xs:string" name="description"
49                      use="optional"/>
50                    </xs:extension>
51                  </xs:simpleContent>
52                </xs:complexType>
53              </xs:element>
54              <xs:element name="Row">
55                <xs:complexType>
56                  <xs:simpleContent>
57                    <xs:extension base="xs:string">
58                      <xs:attribute type="xs:byte" name="name"
59                      use="optional"/>
60                      <xs:attribute type="xs:string" name="description"
61                      use="optional"/>
62                    </xs:extension>
63                  </xs:simpleContent>
64                </xs:complexType>
65              </xs:element>
66              <xs:element name="Column">
67                <xs:complexType>
68                  <xs:simpleContent>
69                    <xs:extension base="xs:string">
70                      <xs:attribute type="xs:string" name="name"
71                       use="optional"/>
72                      <xs:attribute type="xs:string" name="description"
73                      use="optional"/>
74                    </xs:extension>
75                  </xs:simpleContent>
76                </xs:complexType>
77              </xs:element>
78              <xs:element name="Range">
79                <xs:complexType>
80                  <xs:simpleContent>
81                    <xs:extension base="xs:string">
82                      <xs:attribute type="xs:string" name="name"
83                      use="optional"/>
84                      <xs:attribute type="xs:string" name="description"
85                      use="optional"/>
```

```
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element name="Output">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute type="xs:string" name="name"
                      use="optional"/>
                      <xs:attribute type="xs:string" name="description"
                      use="optional"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
            </xs:choice>
            <xs:attribute type="xs:string" name="name" use="optional"/>
            <xs:attribute type="xs:string" name="description"
            use="optional"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute type="xs:string" name="name"/>
      <xs:attribute type="xs:string" name="description"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

APPENDIX B

# PRE-QUESTIONNAIRE

# Questionário Pré-Sessão

Este questionário tem como objetivo selecionar pessoas com alguma experiência no uso de uma ferramenta de edição de folhas de cálculo.

*Obrigatório

1. **Sexo** *
   *Marcar apenas uma oval.*
   - ◯ Masculino
   - ◯ Feminino

2. **Idade** *
   *Marcar apenas uma oval.*
   - ◯ <20
   - ◯ 20-22
   - ◯ 23-25
   - ◯ >25

3. **Curso** *

   .....

4. **Ano curricular em que se encontra** *

   .....

5. **Já trabalhou com alguma ferramenta de edição de folhas de cálculo?** *
   Ex: Excel, OpenOffice, LibreOffice, etc
   *Marcar apenas uma oval.*
   - ◯ Sim
   - ◯ Não *Pare de preencher este formulário.*

6. **Como faria o somatório das células A1 a D3?** *

   .....

7. **Como faria para saber se o valor da célula A1 é maior que o da célula A3?** *

   .....

8. **O que faz a função COUNT?** *

   .....

9. **O que faz função IF?** *

.....

Com tecnologia
Google Forms

Appendix C

# Post-Questionnaire

# Questionário Pós-Sessão

1. Selecione a resposta que corresponde ao quanto concorda ou discorda com as seguintes frases.
   (a) Estou confiante que respondi corretamente a todas as tarefas da folha de cálculo **Adbudget**. (Selecione uma)
      ◯ Concordo plenamente.
      ◯ Concordo.
      ◯ Nem concordo nem discordo.
      ◯ Discordo.
      ◯ Não se aplica.
   (b) Estou confiante que respondi corretamente a todas as tarefas da folha de cálculo **EnronGAS**. (Selecione uma)
      ◯ Concordo plenamente.
      ◯ Concordo.
      ◯ Nem concordo nem discordo.
      ◯ Discordo.
      ◯ Não se aplica.
2. A ferramenta ajudou-o a desempenhar as tarefas propostas com maior facilidade?
      ◯ Sim.
      ◯ Não.
3. Pensa que a ferramenta deveria ajudar mais quando estava a desempenhar as tarefas?
      ◯ Sim.
      ◯ Não.
4. A documentação existente sobre a folha de cálculo é de fácil consulta?
      ◯ Sim.
      ◯ Não.
5. A leitura da documentação existente sobre a folha de cálculo é fácil?
      ◯ Sim.
      ◯ Não.
6. A documentação existente sobre a folha de cálculo é percetível?
      ◯ Sim.
      ◯ Não.
7. O que acha da nossa ferramenta?

8. Sugestoes de melhoria:

Appendix D

# Questions

# Perguntas da folha de cálculo Adbudget

1. Comece por alterar o input fator de gastos em despesas gerais para 1,2.
   Qual é o valor da célula correspondente aos custos das despesas gerais no 3º trimestre?

   Resposta: _____
   Início: Fim:

2. Coloque o input preço unitário do produto a 30.
   Qual é o valor da célula do lucro bruto no 1º trimestre.

   Resposta: _____
   Início: Fim:

3. Indique quais são os inputs (argumentos da fórmula) da célula que calcula as despesas fixas no 4º trimestre.

   Resposta: _____
   Início: Fim:

4. Modifique a fórmula responsável pelo cálculo da receita bruta de unidades vendidas no 1º trimestre, acrescentando-lhe a **multiplicação** do input fator de procura. Escreva a fórmula tal e qual como aparece no Excel.

   Resposta: _____
   Início: Fim:

5. Modifique a fórmula responsável pelo cálculo das despesas fixas no 2º trimestre acrescentando-lhe a **soma** da célula do cálculo dos custos de produção das unidades vendidas no 2º trimestre. Escreva a fórmula tal e qual como aparece no Excel.

   Resposta: _____
   Início: Fim:

# Perguntas da folha de cálculo EnronGas

1. Comece por alterar o input preço do gás no dia 10 correspondente à ENRON - POOL POINT para 12,12.
   Qual é o valor do output valor real do gás na faturação no dia 10?

   Resposta: _____
   Início: Fim:

2. Coloque o input percentagem da quantidade base de gás a 0,9.
   Qual é o resultado do output valor do défice do dia 8.

   Resposta: _____
   Início: Fim:

3. Indique quais são os inputs (argumentos da fórmula) da célula que calcula o volume atual de gás diário no dia 3?

   Resposta: _____
   Início: Fim:

4. Modifique a fórmula responsável pelo cálculo do volume atual de gás diário no dia 3 retirando o input volume atual de gás diário no dia 3 na zona HIGH ISLAND 13L. Escreva a fórmula tal e qual como aparece no Excel.

   Resposta: _____
   Início: Fim:

5. Modifique a fórmula responsável pelo cálculo total do valor da faturação para passar a ser a fórmula AVERAGE. Escreva a fórmula tal e qual como aparece no Excel.

   Resposta: _____
   Início: Fim:

APPENDIX

# E

# TUTORIAL

# Tutorial

## Introdução

No contexto desta investigação produzimos uma ferramenta, designada *SpreadsheetDoc*, que suporta documentação, para uma melhor compreensão das folhas de cálculo.

Na prática, essa ferramenta fornece um conjunto de botões que, cada um, executa um conjunto de operações.

Neste estudo, pretendemos analisar até que ponto é vantajoso utilizar a ferramenta que nós desenvolvemos.

Se tiver alguma dúvida durante este tutorial, por favor diga ao supervisor, para que o mesmo o possa esclarecer.

Este tutorial pretende fornecer um primeiro contacto com a nossa ferramenta, e utiliza uma folha de cálculo para calcular a probabilidade de se comprar o barco SS Kuniang conforme a proposta feita pela empresa. Alterando o valor oferecido, a folha apresenta também o lucro sobre essa oferta.
Abra a folha de cálculo SS Kuniang.xls situada no ambiente de trabalho. De seguida, clique na tab SpreadsheetDoc, situada na barra do menu do Excel. É possível visualizar na Figura 1.

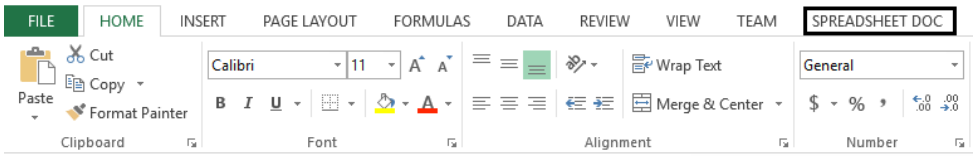



Figure 1: Tab da nossa ferramenta

A ferramenta desenvolvida apresenta 5 grupos de documentação: General Documentation, Content Documentation, Input/Output Documentation, **Read Documentation** e XML Documentation. Cada um destes grupos tem as suas respetivas funcionalidades. **Tenha em atenção, que neste tutorial vamos apenas focar-nos nas funcionalidades do grupo marcado a negrito. É possível visualizar na Figura 2**.

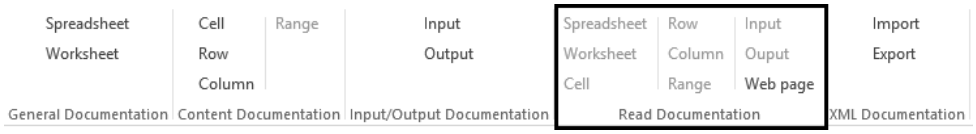



Figure 2: Funcionalidades da ferramenta

Vamos agora abrir a documentação da folha de cálculo numa página web. Vá até ao ambiente de trabalho e abra o ficheiro SS Kuniang.html

A página web começa por mostrar uma descrição geral do funcionamento da folha de cálculo. De seguida, é possível ver todas as worksheets presentes na folha (neste caso Assumptions e Model). Cada worksheet tem respetivamente a sua descrição e o seu conteúdo, ou seja, células, inputs, outputs, colunas, linhas e ranges que estejam documentados. Vamos agora explicar cada uma das secções.

A secção das células contem a lista de todas as células documentadas. No caso da célula conter um valor, então tem apenas a descrição. Caso a célula seja uma fórmula, além da descrição, tem todos os inputs da célula (argumentos da fórmula, cada um com a respetiva descrição) e o output (também com a respetiva descrição).

A secção dos inputs contem a lista de todos os inputs (células) documentados e a respetiva descrição.

A secção dos outputs contem a lista de todos os outputs (células) documentados e a respetiva descrição.

A secção das colunas contem a lista de todas as colunas documentadas e a respetiva descrição.

A secção das linhas contem a lista de todas as linhas documentadas e a respetiva descrição.

A secção dos ranges contem a lista de todos os ranges documentados e a respetiva descrição.

Antes de passarmos às perguntamos vamos explicar como ver os **inputs de uma célula (argumentos da fórmula)**. Para uma célula ter inputs tem que ser uma célula com uma fórmula. Siga os seguintes passos.

1. Selecione a célula I4 na worksheet Model.
2. Clique no botão que diz Cell.
3. Leia os inputs da célula (argumentos da fórmula) e verifique que os inputs são C10, F4, C7 e C8. Como pode ver na Figura 3, os inputs estão marcados com um retângulo preto à volta.
4. Clique no botão que diz Close para fechar a documentação dessa célula.

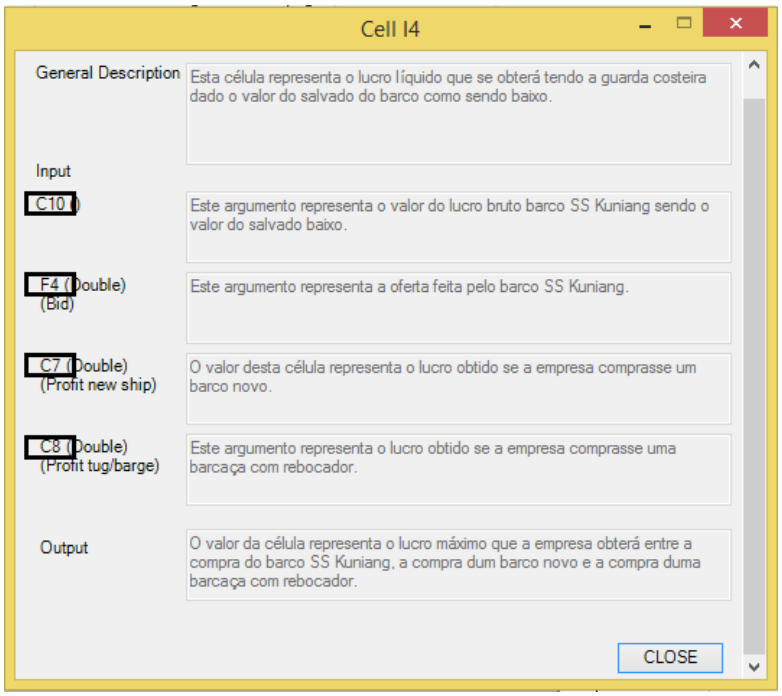



Figure 3: Identificação de inputs de uma célula

**Ao resolver as perguntas propostas tenha sempre em atenção se está a trabalhar na worksheet correta.**

## Perguntas

**Pergunta 1) Comece por alterar o input lucro bruto quando o valor do salvado é baixo para 18,5.**
**Qual é o valor da célula lucro líquido que se obterá tendo a guarda costeira dado o valor do salvado como sendo baixo?**
Responda à pergunta seguindo os seguintes passos.

1. Ler a documentação das duas worksheets na página web.
2. Após a leitura, foi possível perceber que vamos ter que fazer alterações na **worksheet Assumptions** para alterar o input pretendido. Selecione na página web a tab Assumptions.
3. Selecione na folha de cálculo a **worksheet Assumptions**.
4. Como a pergunta diz **input**, vamos procurar na página web na secção dos inputs. Leia a documentação das células de input. Depois de encontrar o input pretendido, neste caso é o C10, selecione na folha de cálculo a célula C10 da **worksheet Assumptions**.
5. Altere o valor da célula C10 para 18,5, tal como diz na pergunta.
6. Como a pergunta pede o valor de outra **célula**, temos de perceber onde ela está. Temos de procurar na documentação a célula pretendida. Leia a documentação das células de ambas as worksheets. Verá que a célula I4 da **worksheet Model** é a que pretendemos.
7. Consulte a célula I4 da **worksheet Model** e escreva o seu valor na resposta. (A resposta encontrada deve ser 6,500).

Resposta: _____

**Perguta 2) Coloque o input oferta feita pelo salvado do barco a 9.**
**Qual é o valor do output lucro que a empresa espera obter com o barco SS Kuniang?**
Responda à pergunta seguindo os seguintes passos.

1. Como a pergunta diz **input**, vamos procurar na página web na secção dos inputs. Leia a documentação das células de input. Depois de encontrar o input pretendido, neste caso é o F4, selecione na folha de cálculo a célula F4 da **worksheet Model**.
2. Altere o valor da célula F4 para 9, tal como diz na pergunta.
3. Como a pergunta pede o valor do **output**, temos de perceber onde ele está. Sendo que apenas a **worksheet Model** produz resultados e outputs, então temos de procurar na documentação o output pretendido. Leia a documentação dos outputs da **worksheet Model**. Verá que a célula F9 da **worksheet Model** é a que pretendemos.
4. Consulte a célula F9 da **worksheet Model** e escreva o seu valor na resposta. (A resposta encontrada deve ser 6,350).

Resposta: _____

**Pergunta 3) Indique quais são as inputs (argumentos da fórmula) da célula do lucro líquido que se obterá tendo a guarda costeira dado o valor do salvado do barco como sendo baixo.**
Responda à pergunta seguindo os seguintes passos.

1. Como na pergunta pede as **inputs da célula (argumentos da fórmula)**, temos que ir procurar na secção das células pela **célula** pretendida. Depois de encontrarmos a célula pretendida, neste caso I4, selecione na folha de cálculo a célula I4 da **worksheet Model**.

2. Clique agora no botão que diz **Cell** para abrir a documentação da célula no próprio Excel.

3. É possível ver que a célula tem quatro inputs.

4. Escreva quais são os inputs da célula na resposta. (A resposta deve ser C10, F4, C7, C8).

Resposta: _____

**Pergunta 4) Modifique a fórmula responsável pelo cálculo da probabilidade de se comprar o barco SS Kuniang para passar a ser uma percentagem, ou seja, multiplique a fórmula existente por 100.**
Responda à pergunta seguindo os seguintes passos.

1. Como na pergunta pede para **modificar uma fórmula**, temos que ir procurar na secção das células e ver qual é a **célula** com a fórmula pretendida. Depois de encontrarmos a célula pretendida, neste caso F5, selecione na folha de cálculo a célula F5 da **worksheet Model**.

2. Acrescente agora à fórmula a multiplicação por 100. A fórmula é =(F4-2)/10

3. Escreva a fórmula tal e qual como está no Excel na resposta. (A resposta deve ser =((F4-2)/10)*100.

Resposta: _____

**Pergunta 5) Modifique a fórmula do valor líquido que a empresa obterá no caso da oferta apresentada não ser suficiente para passar a ser a fórmula MIN.**
Responda à pergunta seguindo os seguintes passos.

1. Como na pergunta pede para **modificar uma fórmula**, temos que ir procurar na secção das células e ver qual é a **célula** com a fórmula pretendida. Depois de encontrarmos a célula pretendida, neste caso I10, selecione na folha de cálculo a célula I10 da **worksheet Model**.

2. Vamos então alterar a fórmula presente, ou seja, passar de MAX para MIN. A fórmula é =MAX(C7;C8).

3. Escreva a fórmula tal e qual como está no Excel na resposta. (A resposta deve ser =MIN(C7;C8).

Resposta: _____