# A methodology for constructing the calculation model of scientific spreadsheets

Martine de Vos*
Computer Science
Network Institute
VU University Amsterdam
The Netherlands
martine.de.vos@vu.nl

Jan Wielemaker
Computer Science
Network Institute
VU University Amsterdam
The Netherlands
j.wielemaker@vu.nl

Guus Schreiber
Computer Science
Network Institute
VU University Amsterdam
The Netherlands
guus.schreiber@vu.nl

Bob Wielinga
Computer Science
Network Institute
VU University Amsterdam
The Netherlands
b.j.wielinga@uva.nl

Jan Top†
Wageningen University and
Research Centre
Food and Biobased Research
The Netherlands
jan.top@wur.nl

## ABSTRACT

Spreadsheets models are frequently used by scientists to analyze research data. These models are typically described in a paper or a report, which serves as single source of information on the underlying research project. As the calculation workflow in these models is not made explicit, readers are not able to fully understand how the research results are calculated, and trace them back to the underlying spreadsheets. This paper proposes a methodology for semi-automatically deriving the calculation workflow underlying a set of spreadsheets. The starting point of our methodology is the cell dependency graph, representing all spreadsheet cells and connections. We automatically aggregate all cells in the graph that represent instances and duplicates of the same quantities, based on analysis of the formula syntax. Subsequently, we use a set of heuristics, incorporating knowledge on spreadsheet design, computational procedures and domain knowledge, to select those quantities, that are relevant for understanding the calculation workflow. We explain and illustrate our methodology by actually applying it on three sets of spreadsheets from existing research projects in the domains of environmental and life science. Results from these case studies show that our constructed calculation models approximate the ground truth calculation workflows, both in terms of content and size, but are not a perfect match.

*Corresponding author

†Second affiliation: Computer Science, Network Institute, VU University Amsterdam, The Netherlands, j.l.top@vu.nl


*KCAP* '15 Palisades, NY, USA


## Categories and Subject Descriptors

I.6 [**Computing methodologies**]: Simulation and Modeling—*Model Validation and Analysis*; J.2 [**Computer Applications**]: Physical sciences and engineering

## General Terms

Algorithms, Human Factors, Verification

## Keywords

Spreadsheets, Calculation Model, Graph Aggregation, Heuristics

## 1. INTRODUCTION

In this article we propose a methodology for semi automatically deriving the calculation workflow from scientific spreadsheet models by aggregating these based on formula syntax and heuristics.

Spreadsheets are one of the main tools used by scientists to store and analyze research data [18, 13]. These scientists typically describe their computational spreadsheet model and corresponding simulations in a paper or a report. This written publication does not serve as documentation, but rather provides readers with a concise explanation of the underlying concepts and an interpretation of the simuation results. Although more and more scientific computational models and associated data become publicly accessible, the models themselves are often too big or too complicated for people to understand easily. In practice, the publication serves as single source of information on the underlying research project. However, it would be desirable if readers of these publications are able to fully understand, i.e., both on a procedural and conceptual level, how the research results are calculated, and trace them back to the underlying spreadsheets. The main problem is that the calculation workflow in the spreadsheets is not made explicit. The goal of this study is to solve this problem by exploring to what extent the calculation workflow underlying a set of spreadsheets can be made explicit.

We distinguish different levels of modeling which all play a different role in our study. First there is the theoretical model that is described in the written publication, and which explains the concepts and relations in the system of interest. The theoretical model is an abstract model that can be described in the text, or presented in a diagram or in an equation, like the differential equation in our first case study on glaciers: $\tau_Y = \tau_D = \rho g H \frac{\delta h}{\delta x}$.

The theoretical model is then translated into procedural statements and calculations, i.e., the calculation model. This model still contains domain concepts and relations, but is formulated in such a way that it enables actual calculation of model results. The calculation model may or may not be described in the written publication. In our glaciers case study the calculation model is represented by a numerically discretized equation, which is the solution to the differential equation. A fragment is shown in the follwing equation: $c = h_i(B_{i+1} - H_i) - \frac{2\Delta x \bar{\tau}_Y}{\rho g}$

Finally, the equations from the calculation model are implemented in spreadsheet syntax and organized in tables, i.e. the computational model. In our glaciers case study this looks like the following formula:
$R11 = (G10 * (C11 - H10)) - ((2 * (B11 - B10) * (((E10 + E11)/2)/F10))/(\$C\$2 * \$C\$3))$

In this paper we propose a methodology to semi automatically derive the calculation model from the computational model. This constructed calculation model provides us with insight on how research results are calculated. Ideally, we would construct the theoretical model, which provide us with more understanding of the domain knowledge included in the calculation workflow, but we expect that this model can not directly be constructed from the computational model.

We consider the computational model as a set of quantities related to each other by spreadsheet formulas. In order to construct the calculation model, we need to determine which of the quantities in the computational model are relevant and how these can be recognized. The starting point of our methodology is the cell dependency graph. In this graph the spreadsheet formulas are abstracted to the in- and output cells, i.e., the nodes, and the dependencies between the cells, i.e., the edges. The cell dependency graph provides a good understanding of the data flow and the structure of the spreadsheets [4, 14].

The basic assumption in our methodology is that spreadsheets contain many redundant cells in the form of instances and duplicates of the same quantities. We automatically aggregate or remove these redundant cells based on the syntax of the formulas. Subsequently, we use a set of heuristics to distinguish important and auxiliary quantities, and to achieve further aggregation of the cell dependency graph. The resulting set of nodes from the fully aggregated graph are labeled and serve as input for the calculation workflow.

In our methodology, we use three different types of knowledge, i.e., 1) knowledge on spreadsheets, i.e., design and syntax, 2) knowledge on computational procedures, and 3) knowledge on the research domain of the spreadsheet models. The first two types are used in the aggregation algorithm, and the heuristics incorporate all three types of knowledge.

We explain an overview and the basic principles of our methodology in section 3. In section 4 and 5 we illustrate our methodology by actually applying it in three case studies, which are all existing research projects in the domains of environmental and life science. For each case study, we compare our results with a manually constructed calculation model created from the equations and descriptions in the corresponding research papers.

# 2. RELATED WORK

Numerous researchers in different subfields of Computer Science and Information Science have developed approaches to support spreadsheet developers during the development and use of their spreadsheets [10]. There are many commercial tools available for end-users, like, for example, Spreadsheet Detective, Excel Auditor, and SLATE. However, these tools are primarily designed for detecting and fixing errors, and are less suitable to support users in understanding the entire calculation workflow within a workbook.

Although most scientific approaches also focus on errors in spreadsheet applications, these may, at the same time, help users to better structure and understand their data. Many of these approaches use visualization techniques to support user understanding, and several specifically visualize the data flow or dependencies within the spreadsheets. Dependencies between different parts of the spreadsheets are, for example, visualized in animations [9], in three-dimensional space [8], or in larger blocks of formulas [16]. Other approaches visualize the dependencies of the entire workbook in a graph like structure [6] and apply visual abstraction techniques to make the graph manageable [11, 7]. But as these techniques only offer visual representations on different abstraction levels, the graph itself remains unchanged and still too complex to be understood.

Clermont and colleagues [5] developed a toolkit for creating abstract representations of spreadsheet programs, to facilitate understanding and checking of complex spreadsheets. They examine the cell dependency graph of the entire workbook, and semi-automatically aggregate these in semantic classes, i.e., groups of spreadsheet cells with similar formula structure, and data modules, i.e., groups of spreadsheet cells that are input to a distinguished result cell.

Unfortunately, as already observed by [14], many of the approaches mentioned above have not been implemented [6, 8] or are currently not available for use by third parties [9, 5]. Although we were not able to try and analyze the corresponding tools ourselves, the approaches provide useful insights on analysis and visualization of formulas and dependencies in spreadsheets. The aggregation and selection of information within the cell dependency graph remains an issue and is the focus of present study.

# 3. METHODOLOGY

## 3.1 Basic principles

In the calculation workflow we want to include quantities that both represent domain knowledge and are essential for understanding the calculation of the research results. We assume that, in a set of spreadsheets, multiple cells refer to the same quantity, as they refer to different instances or copies of that quantity. The number of spreadsheet cells, and consequently the number of nodes and edges in the cell dependency graph, is therefore greater than the number of quantities present in the spreadsheets. We develop an algorithm that recognizes and aggregates these instances and

copies based on analyzing the formula syntax.

We also assume that only part of the quantities in the computational model has a direct contribution to the computation of research results, and is therefore essential in the calculation workflow. The other part consists of auxiliary quantities that support the developers in designing and testing the computational model, and may therefore be left out of the calculation workflow. We use knowledge on spreadsheet design and computational procedures to develop a set of heuristics. We apply the heuristics on the aggregated graph to select the quantities that have a direct contribution, and use these as input for the calculation workflow.

Summarizing, our approach roughly consist of three steps. The starting point of our methodology is the construction of the cell dependency graph (1). Then we aggregate the cell dependency graph based on formula syntax (2), and finally we aggregate the cell dependency graph by applying heuristics (3). We explain and illustrate our methodology by applying it on three case studies. The first two steps are developed independent from these case studies. The heuristics, on the other hand, are developed based on analysis of these case studies.

The different steps will be described in more detail in the next sections. All algorithms are developed using SWI prolog[1] and are publicly available[2].

## 3.2 Construction and aggregation of the cell dependency graph

We automatically generate the cell dependency graph by parsing all formulas in a set of spreadsheets, and analyze the dependencies between the corresponding cells (Algorithm 1).

**Algorithm 1** Create cell dependency graph

```
for all cells do
    for all cells containing a formula do
  ▷ add here aggr step 1 (simplify lookup) and aggr step 2 (formula groups)
        for all cells after "=" sign do
            cell = inputcell
        end for
        create pair: [outputcell - [inputcell1, inputcell2, ...]]
    end for
    collect pairs in CellInputsList: [outputcell1 - [inputs], outputcell2 - [inputs], ... ]
end for
for all inputs in CellInputsList do
    if input NOT outputcell  then
        input = source cell
        create pair:[source cell - [ ] ]
    end if
    NewCellInputsList = CellInputsList + source pairs
    transpose NewCellInputsList into [cell1 - [dependents], cell2 - [dependents], ... ]
end for
```

We aggregate the cell dependency graph by adding three different aggregation steps to the original algorithm (Algorithm 1. First, we simplify formulas with a built-in "LOOKUP" function . These functions can be considered dynamic copy functions, as these functions refer to the value in another spreadsheet cell, but represent the corresponding cell address as a formula. However, as the formula syntax is such that it refers to a whole cell range, these "LOOKUP" functions yield a lot of redundant nodes and edges in the cell dependency graph. If the data that the "LOOKUP" functions are referring to are present, the functions can be solved and replaced by single references to the target cells.

Second, we aggregate copy-equivalent regions [5], i.e., blocks of cells that contain formulas with the same syntax (Algorithm 2). We generalize formulas in order to compare their structure and relative cell positions. Formulas may be aggregated over multiple columns, rows, sheets or rectangular areas. Instead of the original formulas, the list of aggregated formulas is now used to determine the dependencies in the spreadsheets.

And last, we recognize all spreadsheet cells that copy the value of another cell, i.e., the original. In the dependency graph, we replace the corresponding cells with the original cells.

**Algorithm 2** Create formula groups

```
for all cells do
    for all cells containing a formula do
        generalize formula: replace cell coordinates by variables X and Y or constraints thereof
    end for
end for
for all generalized formulas do
    group formulas with the same key, syntax and relative cell positions
    if formulas can be grouped then
        represent group as
        Node = ForAll(What, In, f(Sheet,X,Y,Formula))
        What = one of [row,column,sheet,area], In = range
    else
        single formula
    end if
end for
GroupedFormulaList = single formulas + formula groups
```

## 3.3 Development and application of heuristics

We develop a set of heuristics to distinguish relevant from auxiliary variables in the spreadsheets. Few studies report, in general terms, on design characteristics of spreadsheets. These studies state that spreadsheets usually are a mixture of input data and computations, and contain "checks and balances" to provide visual feedback to the user or developer [1, 8] We combine this knowledge with an in-depth analysis of our three case studies and formulate a set of concrete heuristics.

In this study we apply the heuristics manually, as this this is the most fast and easy option, given the small data set. We formulate the heuristics in such a way that, if they prove succesful, these could eventually be integrated in our algorithms.

After manual application of the heuristics, the nodes in the resulting graph are manually labeled to enable comparison with the ground truth calculation workflow. We follow the method of Hermans and colleagues [7] by labeling the nodes with the textual values that are found in the corresponding table headers. If the nodes represent an aggregation over a range of cells in either a column or row, this range is included in the label either by providing all corresponding header values or by providing the coordinates of the range.

# 4. CASE STUDY: SET UP

[1]SWI-prolog, `http://www.swi-prolog.org/`

[2]Library plsheet `https://github.com/Data2Semantics`

**Table 1: Characteristics of the three case studies**

| | glaciers | hormones | pollutants |
|---|---|---|---|
| # sheets | 1 | 5 | 7 |
| # cells | 525 | 2220 | 2703 |
| # formulas | 312 | 1676 | 1666 |
| formula types | arithmic | arithmic,copy, AVERAGE,IF COUNT,STDEV | arithmic,copy, AVERAGE,MAX COUNT,LARGE HLOOKUP,VAR |

The application of our methodology is illustrated by applying it in three existing scientific research projects that use spreadsheets to perform analyses. We select our case studies by performing a literature search. We select journal papers 1) from the domains of environmental or life science, that 2) have made the spreadsheets publicly available as supplementary data, and 3) contain both textual descriptions and equations that explain the variables and computations in these spreadsheets.

For each case study we manually construct the ground truth calculation workflow from the corresponding journal paper. Subsequently, we semi-automatically generate the set of quantities to be included in the calculation workflow, following the steps in our methodology. We compare the variables from this ground truth with the set of quantities constructed by our methodology, in order to determine whether our methodology is performing according to our expectations.

## 4.1 Raw Data

All case study models contain spreadsheets that are not connected to the other spreadsheets, or do not contribute to the computation of the model results. We remove these spreadsheets from our analyses.

The first case study is based on a spreadsheet model for reconstructing the surface profile of former mountain glaciers and ice caps [2]. We use one sheet, i.e., the advanced version of the model for reconstructing the surface profile of glaciers in. The second case study is a spreadsheet model to calculate free serum concentrations of hormones in humans [12]. In our case study we use the sheets that contain input data, calculations of binding constants, and calculations of free hormone concentrations. The last case study is a spreadsheet model to score dangerous chemical pollutants related to the exposure scenarios of human risk and to evaluate the uncertainty of the scoring procedure [15]. We use seven sheets, i.e., the sheets containing scoring matrices for pollutants, and the sheets collecting the corresponding scoring results.

The glaciers case study contains less content cells and formulas than the other two case studies. Furthermore, the formula cells in the glaciers case only contain arithmic operators, while those in the other case studies also contain copy actions and built-in spreadsheet functions (Table 1).

## 4.2 Construction of the ground truth calculation workflow

The written publications of our case studies each contain a "model" section. In this section the researchers explain how they translate their theoretical model into the quantities, and expressions that are included in the calculation model. We use this calculation model as the ground truth calculation workflow in our study. However, we are aware that this is not a ground truth in the strict sense of the term, as it is our interpretation of the calculation model that is present in the written publication.

We determine which quantities,i.e., variables, and connections are present in the equations in the "model section", and include these as nodes and edges respectively, in a graph. In this process we use two additional rules. As a first rule, we only include quantities that are directly related to the domain knowledge desrcibed in the written publication. The second rule concerns the manifestation, i.e., the abstraction level, of quantities. In their papers the researchers may choose different abstraction levels, of quantities in their calculation model. They may choose to abstract over instances, for example, over individual hormones or pollutants. Or they may explicitly include different abstraction levels of the same quantity in their equations. In the pollutants case, for example, the researchers distinguish a quantity that is related to an individual pollutant from that same quantity that is related to all pollutants. In the construction of our ground truth we follow the abstraction level that is chosen by the researchers.

The written publications may also contain information on how the variables, and equations are implemented in the spreadsheets, i.e., the computational model. We do not use this information in the construction of the ground truth calculation workflow. Yet, this information may be useful in analyzing and explaining the results when comparing the ground truth with the automated calculation workflow.

## 4.3 Results of the ground truth calculation workflow

The results of the ground truth calculation workflows are shown in Table 2 and Figures 1 and **??**. The ground truth of the pollutants case is not shown, as it is too big to display.

The spreadsheet model in the glaciers case study is built on an initial equation describing the driving stress, i.e. the stress that causes a glacier to deform, as a function of the weight and surface gradient of the ice. The model in the hormones case study is based on a set of equilibrium equations describing the concentrations of free hormones and unoccupied protein binding sites. In both case studies the researchers have numerically discretized this equation to derive an approximate solution to the equation, and describe this process in the "model" section of their paper. We use the discretized equations as the ground truth calculation workflows of both case studies.

In the hormones case study, the equations in the model section apply to five individual hormones. We follow the approach of the researchers, by considering these hormones as instances, and interpret the equations as applicable to a single hormone "X" . In the pollutants case study, the written publication contains a separate section on uncertainty evaluation. We do not include this section in the ground truth calculation workflow, as we think the equations in this section describe the procedural aspects of the computations, rather than the domain knowledge included in these computations.

## 4.4 Comparison with the ground truth calculation workflow

We manually analyze the agreement between the nodes of the semi-automatic calculation model and the variables

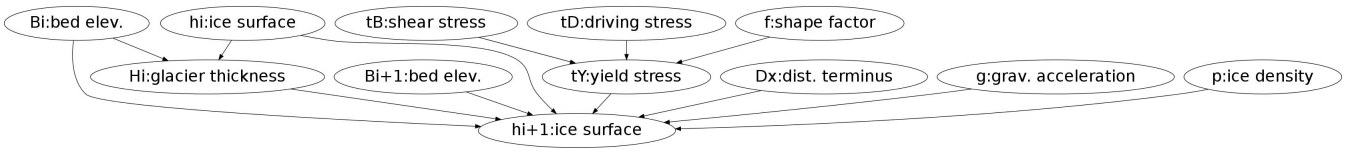


**Figure 1: Manually constructed ground truth calculation workflow of the glacier case study**

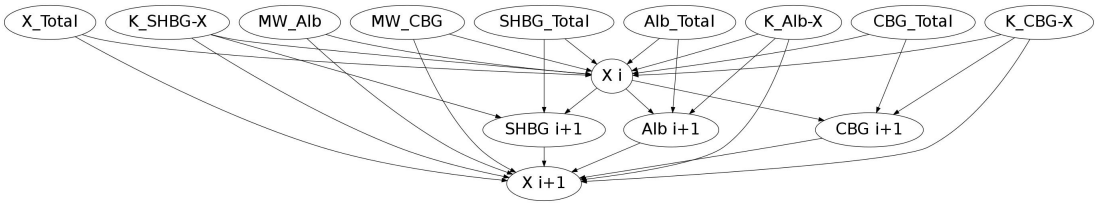


**Figure 2: Manually constructed ground truth calculation workflow of the hormones case study. *X* corresponds to one of the hormones [T,DHT,E2,E1,C]. *ALB, SHBG and CBG* are binding proteins. *K-protein-X* corresponds to the association constant for a particular protein-hormone binding**

**Table 2: Size of the ground truth calculation workflows of the three case studies**

| | # nodes | # edges |
|---|---|---|
| glaciers | 12 | 13 |
| hormones | 18 | 37 |
| pollutants | 90 | 109 |

of the ground truth calculation workflow. We consider the nodes and variables to agree, when the node labels match with the variable names match, and, if applicable, the cell range of the node matches the discretion step of the variable.

We determine the recall by dividing the number of matches by the number of variables present in the ground truth calculation workflow. We determine the precision by dividing the number of matches by the number of variables present in the semi-automatic calculation model.

# 5. CASE STUDY: RESULTS

## 5.1 Automatic aggregation

All aggregation steps reduce the number of nodes and edges in the cell dependency graphs (Table 3), but the grouping of formulas is by far the most effective aggregation step. The total reduction differs per case study and is largest for the glacier case.

For all case studies the number of nodes and edges in the aggregated graph is bigger than the number in the ground truth calculation workflow. The difference in size is most obvious for the hormones and pollutants case studies.

## 5.2 Development and application of heuristics

For all case studies we analyze the potential causes for the difference in size between the aggregated graph and the ground truth calculation workflow and use these as a basis for heuristics to achieve further reduction of the cell dependency graph (Table 4).

We categorize our observations in three main themes, i.e., "disturbed symmetry", "auxiliary calculations" and "unit conversions". The first theme corresponds to the layout of spreadsheet tables. Spreadsheet tables are usually designed such that the content is logically grouped and arranged in symmetric patterns. These patterns are recognized by our algorithm and used in the aggregation process. However, certain elements in spreadsheet tables may disturb the original symmetric patterns. For example, several spreadsheets in the pollutants case contain macro buttons and merged cells, which influence the coordinates of spreadsheet cells as perceived by the algorithm. Another example is that of missing data values in the hormones case. The copy equivalent region is interrupted by these empty spreadsheet cells, which results in an incomplete aggregation.

The second theme covers auxiliary calculations that are present in spreadsheets, i.e., calculations that are not directly related to the domain knowledge, but rather to the calculation procedure itself. These calculations provide the researchers with additional information on the calculation procedure. Usually these calculations are located in separate spreadsheets, or sections, and are not part of the main flow of computations. In the hormones case, for example, the researchers introduced additional calculations to evaluate their iterative approach. And in the pollutants case, the researchers use additional calculations to evaluate the uncertainty of the parameter values and the scoring procedure. A special case, is derivation of values for constants or parameters from empirical data. These calculations may be considered to be related to domain knowledge. However, these calculations usually occur in separate spreadsheets, and only the resulting values are used in the main flow of computations. Therefore, we consider these as auxiliary calculations.

The third theme deals with unit conversions. In many spreadheets it is necessary to convert the unit of measure of variables before these can be used in further calculations. These conversions result in additional variables in the spreadsheet table, which, in fact, still represent the same quantity, and are of no added value to the calculation workflow. For example, the spreadsheets in the hormones case contain variables that represent the concentrations of hormones both in grams, mol and percentages.

We manually apply the heuristics on the case study spreadsheets, and run the aggregation algorithm (Table 5). In both the hormones and the pollutants case study, removing the

**Table 3: Influence of automatic aggregation steps on the size of the cell dependency graph**

| | | glaciers | | hormones | | pollutants | |
|---|---|---|---|---|---|---|---|
| | | # nodes | # edges | # nodes | # edges | # nodes | # edges |
| original | | 504 | 1119 | 1760 | 4828 | 2104 | 2669 |
| separate aggr steps: | remove copies | 503 | 1117 | 1652 | 4720 | 1778 | 2343 |
| | remove lookups | n.a. | n.a. | n.a. | n.a. | 2054 | 2574 |
| | aggregate formulas | 20 | 19 | 227 | 288 | 1045 | 701 |
| all agr steps applied | | 19 | 18 | 211 | 272 | 1032 | 701 |

**Table 4: Heuristics**

| theme | heuristic |
|---|---|
| restore symmetry | remove macro's and merged cells |
| | fill in empty data cells |
| | restore symmetry of quantity lists across instances |
| remove auxiliary calculations | remove (uncertainty) evaluations of internal values |
| | remove (uncertainty) evaluations of internal calculation procedures |
| | remove derivation of constant or parameter values from empirical data |
| remove unit conversions | |

**Table 6: Comparison of nodes and variables in the automated and ground truth calculation workflow**

| | glaciers | hormones | pollutants |
|---|---|---|---|
| # matches | 8 | 8 | 58 |
| Recall | 0.67 | 0.44 | 0.63 |
| Precision | 0.42 | 0.10 | 0.46 |

auxiliary calculations has most influence on the size of the cell dependency graph. None of the heuristics is applicable to the glacier case.

## 5.3 Comparison with the ground truth workflow

We compare the labeled nodes with the variables from the ground truth calculation workflow. The spreadsheet models in both the glacier and the hormones case are implementations of numerically discretized differential equations. In both cases the researchers explain in their paper that the variables in the spreadsheet model are progressively evolving based on the values in the preceding rows, or columns. Therefore we make the assumption that the cell ranges included in the labels of the nodes of the semi-automated graph, can be compared to the discretion steps "i" and "i+1" that are included in the variable names of the ground truth workflow.

In the glaciers and pollutants case, more than 60% of the variables from the ground truth can be found in the semi-automated graph. In these case studies more than 40% of the nodes from the semi-automated graph can be matched with ground truth variables. In the hormones case presision and recall are lower (Table 6).

The results can partly be explained by the way we designed the algorithm for aggregation of the cell dependency graph. In the hormones case the researchers use the unsubscripted hormone "X" in equations as a representative of the five particular hormones instances. Variables from the ground truth containing X should therefore match with nodes from the semi-automated graph containing the aggregation of the five instances. However, in the automated graph there are only nodes related to the individual hormones, and none related to the aggregation. In the spreadsheet table, the five hormones indeed have identical formulas, but the positioning in the spreadsheet table is such that these formulas are not aggregated by our algorithm.

Deficiencies in the aggregation algorithm also cause redundant nodes in both the hormones and the pollutants case. Several nodes in the automated graph span cell ranges that fall within the range of other nodes. These nodes refer to the same quantity, but are counted as separate nodes. Also, the detection of copy statements in the algoritm is incomplete. Futhermore, some nodes show an aggregation of cells that is semantically unlogical, as these nodes refer to multiple quantities at the same time. These nodes are artefacts, but are counted anyway.

Another explanation for our results may be the existence of discrepancies between the ground truth and the spreadsheet implementation. In both the hormones and the glaciers case, the discretization steps of several ground truth variables are indeed reflected in the cell ranges in the labels of the nodes of the automated graph. But for a number of variables, especially in the glaciers case, the cell ranges and discretization steps do not match (Table 7). The discretion step of some variables from the ground truth, like "Dx[i:i+1]:dist. terminus" and "tY[i:i+1]:yield stress" represents an interval, and it is not clear how this should be matched with a cell range in the spreadsheet. The semi-automated graph of the glaciers case contains additional nodes that are used as starting point, for example,'ice surface [8]" and "bed elev. [8]", or as intermediate step, for example, "b" and "c",in the calculation process. The different stages in the calculation are however not included in the ground truth. The semi-automated graph in the hormones case contains protein bound concentrations for three proteins, but in the ground truth only the "Albumine-bound" concentration is mentioned.

In the pollutants case the researchers normalize the values of quantities for individual pollutant by the dividing the mean value for the individual by the maximum value for all pollutants. However, the maximum value is not included as a separate quantity in the spreadsheets, but hidden in normalization formula. In our aggregation procedure, all quan-

**Table 5: Influence of application of heuristics on the size of the aggregated cell dependency graph**

| | | glaciers | | hormones | | pollutants | |
|---|---|---|---|---|---|---|---|
| | | # nodes | # edges | # nodes | # edges | # nodes | # edges |
| auto. aggregated | | 19 | 18 | 211 | 272 | 1032 | 701 |
| separate heuristics: | restore symmetry | n.a. | n.a. | 178 | 247 | 905 | 638 |
| | remove auxiliary calculations | n.a. | n.a. | 118 | 199 | 275 | 196 |
| | remove conversions | n.a. | n.a. | 177 | 231 | | |
| all heuristics applied | | n.a. | n.a. | 84 | 159 | 125 | 87 |

**Table 7: Comparison of variable and node names in the semi-automated and ground truth calculation workflows of the glacier case study**

| ground truth workflow | semi automatic graph |
|---|---|
| p:ice density | ice density |
| g:grav. acceleration | g |
| Bi:bed elev. | bed elev. [8:69] |
| | bed elev. [8] |
| Bi+1:bed elev. | bed elev. [9:70] |
| f:shape factor | F |
| hi:ice surface | ice surface [8:69] |
| | ice surface [8] |
| | ice surface [8:70] |
| hi+1:ice surface | ice surface [9:70] |
| Hi:glacier thickness | H [8:69] |
| | H [8:70] |
| Dx[i:i+1]:dist. terminus | dist. terminus [8:69] |
| | dist. terminus [9:70] |
| tY[i:i+1]:yield stress | shear stress [8:69] |
| | shear stress [9:70] |
| | b |
| | c |
| | step length |

tities are aggregated over all pollutants because of the identical formula syntax. As a consequence, the maximum values present in the spreadsheets can not be distinguished from the values for individual pollutants in the semi-automated graph.

# 6. DISCUSSION

The automatic aggregation procedure is able to reduce size of cell dependency graph in the case studies with 50 to 95%. Our assumption that the existence of multiple instances and copies per quantity causes redundancy in the cell dependency graph appears to be correct. The grouping of formulas based on identical syntax is the most effective aggregation step within the procedure, which indicates that most redundancy is caused by repetion of formulas for multiple studied objects, like hormones or pollutants.

We have identified three themes in the developed set of heuristics, which are each based on a different type of knowledge. The "symmetry" theme is based on knowledge on the layout of spreadsheets. The "auxiliary calculations" theme is based on discomposition of computational procedures in spreadsheets. The heuristic on "unit conversions" is based on domain knowledge. The application of these heuristics results in an additional reduction of the aggregated cell dependency graph by 40 to 85%. None of the heuristics is applicable to glacier case. A possible explanation is that the computational model of the glacier case is small or simple enough for the developers to do without additional design, or decomposition.

The precision and recall from our case studies show that the application of our aggregation algorithm combined with the heuristics seem to approximate the ground truth An important explanation for not having a perfect match is the way we designed our aggregation algorithm. The comparison with the ground truth shows that deficiencies in the algorithm leave redundancy in the semi-automated graph. Some of these deficiencies are difficult to solve, as they involve choices on the conceptual level with unexpected or undesired consequences. For example, we have chosen not to remove nodes with overlapping cell ranges from the aggregated graph. This may result in redundant nodes, referring to the same quantity. However, sometimes these type of nodes are not redundant, for example, in the glaciers and hormones cases, where these represent quantities with different discretization steps. Furthermore, the design of spreadsheet tables may be too complex for our aggregation to recognize and aggregate repeating patterns.

At the same time, the redundancy in the semi-automated graph indicates feasible opportunities for improving on the aggregation procedure, and consequently on the precision of our methodology. And most important, it also indicates that many redundant nodes in the constructed graph are rather a product of imperfect aggregation, than auxiliary quantities that are missed by the heuristics. We therefore believe that our assumption that part of the quantities in the computational model are auxiliary quantities is correct.

We base the selection of our case studies on a sufficient description of the calculation model, including mathematical equations. However, in all case studies the researchers include descriptions and equations of different levels of modeling in the same "model section". Furthermore, some variables and relations from the calculation model are only textually described and not explicitly included in equations. The construction of the ground truth calculation workflow from the written publication, therefore, appeared not at all trivial, and involved many choices and interpretations.

# 7. CONCLUSION AND FUTURE WORK

The goal of this study is to explore to what extent the calculation workflow underlying a set of spreadsheets can be made explicit. The results of this study show that the cell dependency graph of the computational model mainly consists of redundant nodes, especially of multiple instances per quantity, and auxiliary quantities. The presented methodology is able to reduce this redundancy by 95% or more, which brings the size within limits of human visual comprehension. Besides, the heuristics developed in this study appear suitable to distinguish relevant and auxiliary quantities in the

graph. However, despite the substantial reduction of the cell dependency graph, the constructed calculation model approximates the ground truth calculation workflow, both in terms of content and size, but is not a perfect match.

In future work, we are planning to test and improve our automatic aggregation procedure by applying it on more case studies. Another interesting direction for future work, would be the use of domain knowledge in the form of ontologies, as we expect that this could facilitate our methodology in several ways. An ontology on unit of measures and quantities, like OM [13], could facilitate automatic recognition of unit conversions. Ontologies could also support automatic recognition of domain concepts in spreadsheets. This could provide an additional step in the aggregation procedure, and could also facilitate automatic separation of relevant and auxiliary quantities.

Finally, in future work we may experiment with a different set up of our methodology and a different set of heuristics. A promosing direction is the approach of Clermont and colleagues to aggregate the cell dependency graph based on the data flow [5]. Furthermore, in this study we have chosen a bottom-up approach of step by step reducing the original cell dependency graph. Instead, the aggregation could be set up in a more drastic way, by applying a top-down approach. This includes, for example, applying heuristics from the very beginning to classify and select different parts of spreadsheets to be included in the aggregation.

## Acknowledgments

This publication was supported by Dutch national program COMMIT. We thank the anonymous reviewers for their useful comments on our manuscript.

## 8. REFERENCES

[1] R. Abraham and M. Erwig. Inferring Templates from Spreadsheets. In *Proceedings of the 28th international conference on Software engineering.*, pages 182–191. ACM, 2006.

[2] D. I. Benn and N. R. J. Hulton. An Excel spreadsheet program for reconstructing the surface profile of former mountain glaciers and ice caps. *Computers and Geosciences*, 36(5):605–610, 2010.

[3] G. Boulton, M. Rawlins, P. Vallance, and M. Walport. Science as a public enterprise: the case for open data. *Lancet*, 377(9778):1633–5, May 2011.

[4] Y. Chen and H. C. Chan. Visual checking of spreadsheets. In *Proceedings of the European Spreadsheet Risks Interest Group 1st Annual Conference*, pages 75–85, London, 2000.

[5] M. Clermont. A Toolkit for Scalable Spreadsheet Visualization. In *Proceedings of EuSpRIG 2004 Conference*, pages 1–12. European Spreadsheet Risks Interest Group, 2004.

[6] J. . S. Davis. Tools for spreadsheet auditing. *International Journal of Human-Computer Studies*, 45:429–442, 1996.

[7] F. Hermans, M. Pinzger, and A. V. Deursen. Supporting Professional Spreadsheet Users by Generating Leveled Dataflow Diagrams. In *Proceedings of the 33rd International Conference on Software Engineering.* ACM, 2011.

[8] K. Hodnigg, R. T. Mittermeir, and I. Informatik-systeme. Metrics-Based Spreadsheet Visualization Support for Focused Maintenance. In *Proceedings of the European Spreadsheet Risks Interest Group 9th Annual Conference*, pages 79–94, London, 2008.

[9] T. Igarashi, J. Mackinlay, B.-W. Chang, and P. Zellweger. Fluid Visualization of Spreadsheet Structures. In *Proceedings of the IEEE Symposium on Visual Languages*, Halifax, NS, Canada, 1998.

[10] D. Jannach, T. Schmitz, B. Hofer, and F. Wotawa. Avoiding , Finding and Fixing Spreadsheet Errors - A Survey of Automated Approaches for Spreadsheet QA. *Journal of Systems and Software*, pages 1–69, 2014.

[11] B. Kankuzi and Y. Ayalew. An End-User Oriented Graph-Based Visualization for Spreadsheets. In *Proceedings of the 4th International Workshop on End-User Software Engineering*, pages 86–90, Leipzig,Germany, 2008.

[12] N. a. Mazer. A novel spreadsheet method for calculating the free serum concentrations of testosterone, dihydrotestosterone, estradiol, estrone and cortisol: With illustrative examples from male and female populations. *Steroids*, 74(6):512–519, 2009.

[13] H. Rijgersberg, M. Wigham, and J. Top. How semantics can improve engineering processes: A case of units of measure and quantities. *Advanced Engineering Informatics*, 25(2):276–287, Apr. 2011.

[14] S. Roy and F. Hermans. Dependence Tracing Techniques for Spreadsheets : An Investigation. In *Software Engineering Methods in Spreadsheets*, pages 1–4, 2014.

[15] B. Ruggeri. Chemicals exposure: Scoring procedure and uncertainty propagation in scenario selection for risk analysis. *Chemosphere*, 77(3):330–338, 2009.

[16] J. Sajaniemi. Modeling Spreadsheet Audit : A Rigorous Approach to Automatic Visualization. *Journal of Visual Languages & Computing*, 11:49–82, 2000.

[17] H. Shiozawa, K. Okada, and Y. Matsushita. 3D Interactive Visualization for Inter-Cell Dependencies of Spreadsheets. In *Proceedings of the IEEE Symposium on Information Visualization*, an Francisco, CA, USA, 1999.

[18] K. Wolstencroft, S. Owen, M. Horridge, O. Krebs, W. Mueller, J. L. Snoep, F. du Preez, and C. Goble. RightField: embedding ontology annotation in spreadsheets. *Bioinformatics (Oxford, England)*, 27(14):2021–2, July 2011.