# Combining Spatial and Semantic Label Analysis*

Chris Chambers
School of EECS
Oregon State University

Martin Erwig
School of EECS
Oregon State University

## Abstract

*Labels in spreadsheets can be exploited for finding errors in spreadsheet formulas. Previous approaches have either used the positional information of labels or their interpretation as dimension for checking the consistency of formulas.*

*In this paper we demonstrate how these two approaches can be combined. We have formalized a combined reasoning system and have implemented a corresponding prototype system. We have evaluated the system on the EUSES spreadsheet corpus. The evaluation has demonstrated that adding a syntactic, spatial analysis to a dimension inference can significantly improve the rate of detected errors.*

## 1. Introduction

Spreadsheets are widely used [11] end-user programs that contain many errors [10]. To improve the quality of spreadsheets a variety of approaches to prevent, detect, and remove errors from spreadsheet have been investigated. Since preventive approaches, in principle, have to interfere with the spreadsheet creation process that makes spreadsheets so attractive to end users, much research has focused rather on the detection and removal of errors.

One type of error that can be detected in spreadsheets are dimension errors, which occur when units of measurement are used incorrectly in formulas. Units of measurements can be employed as a concrete notion of types that is well known among end users [4], and are used to characterize different kinds of values, much like traditional, more abstract, type systems used in general-purpose programming languages. For example, a floating point number, which has just one type, can nevertheless represent different kinds of quantities, such as length or time values.

Several systems [5, 6, 4] have been developed in order to deal with unit of measurement errors. Among these dimension inference [5] is a method that can be used to automatically find dimension errors in spreadsheets. This approach has been shown to work reliably and effectively in many cases, however it does not take full advantage of the information provided in the spreadsheet as it does not utilize the structure of the spreadsheet and focuses on ensuring that formulas are dimension correct.

In contrast, there are several systems that are designed to directly take advantage of the labels and the structure of spreadsheets. These purely label-based approaches, such as UCheck [2] or the system described in [3], are designed to find formula errors caused by inconsistent label usage. This technique operates in two distinct analysis phases. The first phase defines header or label information for the entire spreadsheet. UCheck is able to infer this while most others require users to annotate the labels for every cell. Once the headers are determined for a sheet, labels are assigned to cells based on headers and formulas. In the second phase this information is analyzed to find errors.

One thing to note when looking at label analysis and dimension analysis is that both systems rely on header and label information, however, how this information is used is quite different. By combining dimension analysis with the purely label-based approaches the structure of the spreadsheet could be used to help strengthen the reasoning of the system. To some degree this was already tried in the SLATE approach [6]. However, SLATE only transforms labels and dimensions and does not identify errors. Moreover, the fact that SLATE is a stand-alone spreadsheet system that cannot be integrated into Excel and the time it takes for users to annotate a spreadsheet renders the approach currently impractical.

In this paper we describe a way to combine label based reasoning with dimension inference. This approach is achieved by gathering both label and dimension information about a cell. In many cases a spreadsheet contains dimensions on only one axis, while the other axis typically contains labels that do not map to any dimension. These labels, which go unused in dimension inference can help to provide structural information that can be exploited by the reasoning system behind UCheck.

The rest of this paper is structured as follows. In Section 2 we illustrate the issues involved in the adding label reasoning to dimension inference with a small example. In Section 3 we formalize spreadsheets and present models of dimensions and labels. The combined analysis method is described in Section 4. In Section 5 we report on an evaluation of a prototypical implementation of a tool for dimen-

*This work is partially supported by the National Science Foundation under the grant ITR-0325273 and by the EUSES Consortium (`http://EUSESconsortium.org`).

| | A | B | C | D | E | F | |
|---|---|---|---|---|---|---|---|
| 1 | Car | Miles | Gallons | MPG | Total Gallons | Miles Possible | |
| 2 | VW Bug | 180 | 5 | =B2/C2 | 12 | =E2*D3 | |
| 3 | Camry | 225 | 5 | =B3/C3 | 10 | =E3*D3 | |
| 4 | BMW | 300 | 5 | =B4+C4 | 15 | =E4*D4 | |
| 5 | | =SUM(B2:B4) | | | | =MAX(F2:F4) | |
| 6 | | | | | | | |

**Figure 1. Example spreadsheet**

sion analysis. We discuss related work in Section 6 and give conclusions in Section 7.

## 2. An Example

To explain how the integration of spatial and semantic label analysis works, we will show how both dimension inference and the integrated system work on the spreadsheet in Figure 1. This spreadsheet is calculating how far specific cars can travel on a full tank of gas. This data is correlated based on the result of a drive using only five gallons of gas.

When the spreadsheet is checked with dimension inference, it would first identify the headers for all cells. When the headers are analyzed for dimension information, B1, C1, D1, E1, and F1 all map to a valid dimension. This would allow the system to check all formulas in this spreadsheet for dimension correctness. In this case, the system would detect that there is an error in cell D4 where the formula is trying to add miles and gallons.

Upon further inspection it could be noted that the spreadsheet contains another error. In this particular example, the cell F2 has the formula E2*D3. The dimension for E2 is Gallons, and the dimension for D3 is Miles per Gallon, which, when multiplied together result in the dimension Miles. This result contains no dimension errors, but it doesn't seem right. E3 is actually total Gallons for the Camry, while D2 is the MPG for the VW Bug. Logically, the result does not make sense, however, plain dimension inference would have no way to catch this.

By integrating label reasoning and checking that formulas are both dimension *and* label correct, the system presented in this paper is able to identify a previously unnoticed error. The first step is to determine which header axis (row or column) will be used as the dimension axis. In this case there are several dimensions on the horizontal axis (row 1), but dimensionless labels on the vertical axis (column A). The system then identifies labels and dimensions for each cell. For example, the cells in row 2 would have the label "VW Bug".

With this information assigned, the system can then check to ensure that formulas are dimension and label correct. When the system checks the formula in F2 it can identify that it is multiplying a cell, E2, with the unit Gallons and the label "VW Bug" with the cell D2, which has the unit Miles / Gallon and the label "Camry". While the dimensions work out in this formula, the system will identify an inconsistency with the labels and be able to report this to the user.

## 3. Syntactic Representation

In this section, we will formalize the notions of spreadsheets, dimensions, and labels as a preparation for the formal rule system given in Section 3.4.

### 3.1 Spreadsheets

Spreadsheets ($S$) are functions that is a map addresses ($a \in A$) to expressions ($e$), in particular, $S(a)$ yields the expression stored at address $a$ in the spreadsheet $S$. Expressions can be values ($v$) or references to other cells ($\uparrow a$), or are constructed using arithmetic ($+$ or $*$), aggregating (**count**), or conditional operators.

$$e \quad ::= \quad v \mid \uparrow a \mid e + e \mid e * e \mid \textbf{count}(e, \ldots, e) \mid \textbf{if}(e, e, e)$$

The operations $+$ and $*$ represent, respectively, a whole class of additive operators (including $-$ and MAX) and multiplicative operators (including $/$).

### 3.2 Dimensions

A dimension ($d$) is given by a set of dimension components ($c$). Each component is given by a base ($b$), a conversion factor ($f$), and an integer exponent ($n$). A dimension component can also be a dimension variable ($\delta$). The *identity dimension* $\{\}$ is used for dimensionless values.

$$\begin{array}{rcl} d & ::= & \{c, \ldots, c\} \\ c & ::= & b_f^n \mid \delta \end{array}$$

For each base dimension we identify a default unit with factor 1. For example, the default for length is meter (m), that is, $\mathrm{m} = \mathrm{length}_1^1$, which also means that $\mathrm{cm} = \mathrm{length}_{0.01}^1$ and $\mathrm{ft} = \mathrm{length}_{0.3}^1$. In general, the following relationship holds (where $x$ is a dimensionless number and $b$ is an arbitrary base): $x\, b_f^n = x f\, b_1^n$.

In general, the choice of dimensions is arbitrary and depends on the application. For the task of analyzing dimensions in arbitrary spreadsheets, we have chosen the seven SI units and some further units that we have found in the EUSES spreadsheet corpus [8].

A more detailed discussion of dimensions can be found in [5]

### 3.3 Labels

The labeling structure in this integrated system is a simplified version of the formal model presented in [7]. In particular, since labels will be used only for one axis, we can omit the concept of AND labels, which leads to much simplified rules for combining labels.

The syntax that we use for labels is shown below. (Note the difference between an OR-label $\ell_1 \mid \ell_2$ and the vertical bar | to separate grammar alternatives.)

$$\ell \quad ::= \quad v \mid \ell_1[\ell_2] \mid \ell_1 \mid \ell_2 \mid 1$$

To show the different possible types of labels we will look at several cells in Figure 1. A chain of labels, $\ell_1[\ell_2]$, represents cells that may have a second level label. In this example, the cell B3 has the header A3, which contains the label Camry. Since A3 has as its header the cell A1 with label Car, the label associated with B3 is Car[Camry].

An OR label is used when cells with labels are added together. In general, when two cells are added together, their labels are ORed to produce a resulting label. The cell B5 is a SUM, which adds the three value cells in column B. The three labels used in this formula are Car[VW Bug], Car[Camry], and Car[BMW]. The resulting label is Car[VW Bug] | Car[Camry] | Car[BMW]. Since OR distributes over label chains [7], we can factor this expression to Car[VW Bug | Camry | BMW]. Since the OR expression contains all the first-level labels for the second-level label Car, this label expression can be generalized to Car.

The inference rules for some operations place equality or generalization constraints on the labels allowed for arguments. These constraints are embodied in a label simplification operation $\ell \wr \ell \to \ell$, which is defined as follows.

$$\begin{aligned} \ell \wr \ell &\to \ell \\ \ell \wr \ell[\ell_1] &\to \ell \\ \ell[\ell_1] \wr \ell &\to \ell \\ \ell[\ell_1] \wr \ell[\ell_2] &\to \ell \end{aligned}$$

This operation works only for specific arguments, and if it fails in the premise of an inference rule, this corresponds to the identification of a label error.

### 3.4 On Semantic vs. Syntactic Label Analysis

Before we describe our integrated reasoning tool, we want to point out a principal difference of the two underlying approaches, because even though both UCheck and dimension inference uses labels to determine errors, they use this information in quite different ways.

UCheck essentially exploits the relative position of labels, but it doesn't actually interpret labels. This means that we can rename labels without changing the functionality of the system (at least if the renaming is done systematically). For example, the labels Camry or BMW have no meaning to the system and could be replaced by any other strings. It is how these labels are combined with other labels in formulas that forms the basis of error detection.

Dimension inference, on the other hand, derives some meaning from labels. Instead of treating labels as simple strings, labels in dimension inference actually are interpreted to have some extrinsic semantics. This means that renaming a label could cause errors in a spreadsheet. Using the sheet in Figure 1, if the label in cell B1 is renamed to "KM" the formulas in column D will not be dimension correct as the label for that column is "Miles / Gallon", but the resulting dimension is "KM / Gallons". To keep the formula dimension correct the labels in D1 and F1 would have to be renamed as well.

## 4. Integrated Label and Dimension Analysis

The combined label and dimension analysis of a spreadsheet goes through five distinct steps. The last step applies only in those cases when the fourth step produces underspecified dimensions, that is, when it results in inferred dimensions that contain dimension variables.

1. Header inference
2. Label analysis
3. Identify dimension and label axes
4. Dimension inference
5. Dimension instantiation

Header inference, label analysis, and dimension instantiation are components that we have adopted unchanged from our previous work [5], and they are therefore only briefly described here. Steps 3 and 4 will be described in greater detail in the following.

### 4.1 Header Inference

Header inference analyzes the structure of a spreadsheet and returns a set of headers for each cell. A header is simply the address of another cell. Therefore, header inference produces a binary relation $H \subseteq A \times A$ such that $(a, a') \in H$ says that $a'$ is a header of $a$. In general, one cell can be a header for many cells, and any particular cell can have zero, one, or more headers. For example, cell B1 in Figure 1 is a header for B2, B3, and B4, that is, $H^{-1}(\mathrm{B1}) = \{\mathrm{B2}, \mathrm{B3}, \mathrm{B4}\}$, and A2 and B1 are headers of B2, that is, $H(\mathrm{B2}) = \{\mathrm{A2}, \mathrm{B1}\}$. Header inference essentially works by analyzing the spatial relationships between different kinds of formulas. It can

also take into account layout information. Techniques for header inference have been described in detail elsewhere [1, 2]. In the context of this paper we simply reuse those techniques.

### 4.2 Label Analysis

In the second phase of the integrated system we try to derive a dimension for each label contained in a cell that has been identified as a header by header inference. This process works by (a) splitting labels into separate words, (b) removing word inflections, (c) mapping word stems to dimensions, and (d) combining dimensions into one dimension.

For example, cell E1 in Figure 1 is a header cell and is therefore subject to label analysis. Its value can be split into the two words "Total" and "Gallons", and the plural of "Gallons" can be removed. The resulting "Gallon" can then be mapped to the dimension gal. In contrast, "Total" cannot be mapped into any dimension and will thus be mapped to $\{\}$. Finally, the combination of both dimensions yields "Gallons". If no part of a header label can be mapped to a dimension other than $\{\}$, the label is mapped to a dimension variable $\delta$, which indicates that the dimension is at this time unknown.

### 4.3 Identify Dimension and Label Axes

The goal of our system is to exploit one axis for dimension checking and the other for label checking. In the formal rule system, this separation of analysis is reflected by two new judgments, $S, L \vdash a : \ell$ and $S, D \vdash a : d$. These judgments specify how the integrated system gets labels and dimensions for each cell. The header relationship $H$ identified in the header inference phase has to be partitioned into two parts $H = L \cup D$ where:

- $L$ is the set of headers that define labels
- $D$ is the set of headers that define dimensions

If a cell has two headers, one of which defines dimensions and the other which defines a plain label, both of these pieces of information are exploited to make the inference stronger, as can be seen in the COMBOHDR rule in Figure 2.

To facilitate the partitioning of the header relationship, we have to identify table regions in a spreadsheet and for each table its horizontal and vertical label axes. The information provided by label analysis allows us to the identify the following three cases for axes:

1. No Dimension Axis
2. One Dimension Axis
3. Two Dimension Axes

In the following we will describe each of the possibilities in some detail.

**No Dimension Axis** In the simplest case, that is a spreadsheet that has no units of measurement, there will be no dimension axis. If this is the case, then there is no need to run dimension inference at all. The previous system would not have been able to detect any errors. However, with the integration of label-based reasoning we can run UCheck to detect label errors. While this is not the goal of the system, it does give the tool a function for spreadsheets without dimensions.

**One Dimension Axis** The situation where this system is most useful is when there is one axis that contains dimensions in a spreadsheet. If this is the case, the system will be able to combine label-based reasoning with dimension inference. The system will use the identified dimension and label axes to assign labels and dimensions for each cell in a spreadsheet. This information can then be used to detect errors.

As an example, we again look at Figure 1. The two identified axes will be row 1 and column A. In this case these share a cell, A1, which has been identified as a header for A2, A3, and A4 (for details, see [1]). As it is a header for all of the cells in the vertical axis, it is included with them and ignored in the horizontal axis. To correctly identify the dimension axis, the headers are mapped to a dimension. In this example, every header in row 1 and none of the headers in column A maps to a dimension. This makes the decision easy, and the horizontal axis, row 1, is chosen as the dimension axis, with column A being used for labels.

The labels for all headers in the dimension axis are defined as the one unit, 1, and the dimensions for all headers in the label axis are defined as the unit dimension, $\{\}$. This will give the system the flexibility to use both label-based reasoning and dimension inference.

**Two Dimension Axes** The final case occurs when dimensions exist in both axes. Should this arise the system will not attempt to use label-based reasoning to assist dimension inference. It will instead do a pure dimension analysis using the method described previously.

### 4.4 Combined Label and Dimension Inference

The fourth step of the integrated system is a "label-aware" dimension inference, which inspects each cell containing a formula and derives for it a dimension and a label using the system of rules given in Figure 2. In the previous incarnation of this system a rule application could only result when a dimension could not be inferred. Now an additional failure point has been added. If the labels cannot be combined as described in Section 3.3, then the formula is also declared erroneous.

The relationship between formulas and dimensions is formalized through the following judgments that tie to-

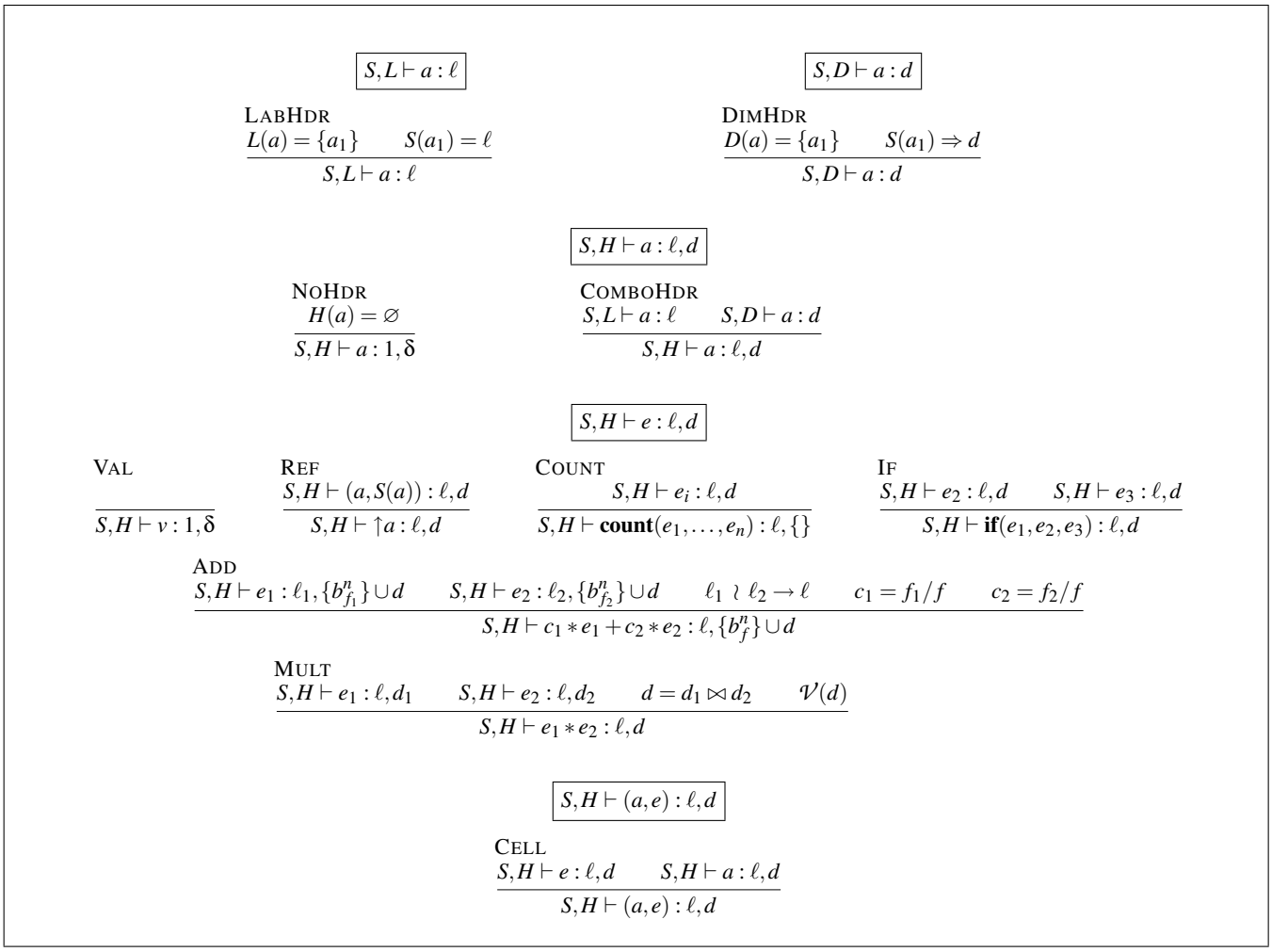


**Figure 2. Combined System rules**

gether the idea of dimension and label axes and the previous judgments from dimension inference.

1. Value Judgment
   $v \Rightarrow d$ says the value $v$, if used as a label or factor, describes the dimension $d$.
2. Header Judgments
   $S,L \vdash a : \ell$ and $S,D \vdash a : d$ transform the header information into dimension and label assignments for addresses. These judgments rely on the separation of $H$ into $L$ and $D$ performed by the dimension/label axis identification step. Specifically, $S,L \vdash a : \ell$ ($S,D \vdash a : d$) says that in the spreadsheet $S$ and given label (dimension) header $L$ ($D$), the location given by address $a$ has dimension $d$ (label $\ell$).
3. Expression Judgment
   $S,H \vdash e : \ell,d$ says that in the spreadsheet $S$ and given the header structure $H$, the expression $e$ has label $\ell$ and dimension $d$.
4. Cell Judgment
   $S,H \vdash (a,e) : \ell,d$ says the cell $(a,e)$ in the spreadsheet $S$ has the label $\ell$ and dimension $d$ under the given header relationship $H$.

To show how these rules are applied to a spreadsheet, we will investigate how they are used on the cells containing errors in Figure 1.

The first such cell is D4. This cell contains an addition and thus will be checked by the ADD rule, which instantiates as follows for D4.

$$\frac{\begin{array}{c} S,H \vdash \text{B4} : \text{Car[BMW]},\{\text{Miles}\} \cup \{\} \\ S,H \vdash \text{C4} : \text{Car[BMW]},\{\text{Gallons}\} \cup \{\} \\ \text{Car[BMW]} \wr \text{Car[BMW]} \to \text{Car[BMW]} \\ c_1 = f_1/f \qquad c_2 = f_2/f \end{array}}{S,H \vdash \text{B4} + \text{C4} : \text{BMW},?}$$

Since the dimensions in this formula are not compatible,

a result dimension cannot be derived, and an error is produced.

The behavior of multiplication errors can be seen by looking at the cell F2, for which the rule MULT is employed.

$$\frac{\begin{array}{c} S,H \vdash \text{E2} : \text{Car[VW Bug]}, \text{Gallons} \\ S,H \vdash \text{D3} : \text{Car[Camry]}, \text{MPG} \\ d = \text{Miles} \qquad \mathcal{V}(\text{Miles}) \end{array}}{S,H \vdash \text{E2} * \text{D3} : ?, \text{Miles}}$$

While the dimensions in this multiplication are fine, the labels Car[VW Bug] and Car[Camry] are not compatible. Thus an error has been identified. For this formula to be correct D3 would have to be changed to D2, which also has the label Car[VW Bug].

### 4.5 Dimension Instantiation

An inferred dimension might contain dimension variables and/or conversion-factor variables. The occurrence of variables happens whenever the spreadsheet doesn't provide enough information to precisely narrow down the dimensions. In these cases we have to find substitutions for the variables to obtain proper dimensions.

The instantiation of dimensions can be realized by generating substitutions for conversion-factor variables so that default dimensions are obtained and by generating substitutions for dimension variables that produce valid dimensions. Of those valid dimensions we can then select the one that is most common (as indicated by the numbers to be reported in Section 5).

## 5. Evaluation

We have extended the dimension inference tool with the ability to perform automatic dimension analysis with label-based reasoning. This tool reuses the header analysis implementation [1] of the UCheck tool [2] and is based on the dimension inference tool in [5]. In this section we describe an evaluation of this system to answer the following research questions.

**RQ1:** *How does the combined system compare to pure dimension inference?*
Dimension inference can be an effective tool to check formulas and spot errors in spreadsheet computations, but it ignores much of the structure of spreadsheets. By integrating label analysis we expect to see a considerable improvement in the detection of errors.

**RQ2:** *Do we lose anything by adding label-based reasoning?*
By integrating label-based reasoning with dimensions it is possible that the system could lose the ability to detect dimension errors. Due to how dimension axes are treated this seems unlikely, but is an important question to answer.

**RQ3:** *By turning off labels on one axis do we lose errors?*
Most label-based systems involve using labels from both axes in a spreadsheet. With one axis being used to define dimensions, how many label errors are not caught by the system.

**RQ4:** *How many tables map cleanly into two axes?*
If the tables in a spreadsheet do not map cleanly into two axes, it will be difficult for the integrated system to run successfully. Knowing how many tables do not map cleanly will provide a negative estimate of how useful the system can be.

**RQ5:** *How many tables contain dimensions on only one axis?*
The primary case for this tool is when dimensions occur on only one axis of a table. We will try to determine exactly how many spreadsheets fall into the category of having one dimension axis and one label axis. This will allow us to further gauge the usefulness of this tool.

### 5.1 Experiments

To answer these research questions we have employed the EUSES spreadsheet corpus [9], which currently contains 4498 spreadsheets collected from various sources. Of these 4498 spreadsheets only 487 contain both dimensions and formulas. Dimension inference, the integrated version, and UCheck were ran on all of these spreadsheets to determine how well each system fared and how many errors each found.

For RQ1 and RQ2, we compared the results of dimension inference and the integrated system. Specifically, how many additional errors did the new system find that were not able to be caught by the old system, and how many errors were not able to be caught when label checking is introduced.

To investigate RQ3 we compared the results of the new system with those of UCheck. By doing this comparison we were able to identify possible label errors that the integrated system was not able to detect, but that were caught when both axes are used for a pure label-based system.

Finally, for RQ4 and RQ5 the mapping of axes for all the 487 spreadsheets with dimensions was investigated to see how many mapped cleanly and to how many dimension axes. Since spreadsheets can have multiple tables, these mapping is performed on all 567 tables in these spreadsheets.

### 5.2 Results

The main experiment was running the three systems on the subset of dimension sheets in the EUSES corpus. This process gave us the necessary information to be able to determine how well the new system operates.

Dimension inference was able to detect 47 spreadsheets with a total of 241 dimension errors, the integrated system

was able to detect 77 spreadsheets with errors and a total of 674 errors, and UCheck identified errors in 17 sheets with 151 total errors. The overlap of errors is shown in Figure 3.

To determine false positives, the spreadsheets with errors were looked at closer. We found that the integrated system produced 11 false positive instances that resulted in 78 total errors, whereas dimension inference found 7 false positive instances that resulted in 49 total errors. The 7 false positives of dimension inference were all inherited by the integrated system. Of the 7 false positives, 6 were caused by label analysis, and one was caused by incorrect header inference. Of the additional 4 false positives in the integrated system one was caused by label analysis and 3 by incorrect header inference.

To determine how many tables contain dimensions in both axes, the axes mapping was logged for every relevant table. When this mapping was performed 128 out of a total of 567 tables contained dimensions in both axes.

There were also 22 tables that did not map well into axes. In general, these were caused by a failure in header inference, in which the headers were not able to be determined.

### 5.3 Discussion

Due to how the new system handles the axes in a spreadsheet the new system did not "lose" any dimension errors. All of the dimension errors are caught by both systems. This is unsurprising as label integration in no way changes how dimensions are handled. If both axes contain dimensions then the system will ignore label-based reasoning and simply perform dimension inference. If dimensions only occur on one axis, then labels will not reduce the number of errors that can be detected.

The final comparison brings up one shortcoming of the system. When we look at the results from UCheck and the results from the new system, there are 6 spreadsheets and 99 additional label errors that the new system was not able to catch. These errors all are caused by a label error based on the dimension axis. Since that axis was being used to define the dimensions in the spreadsheet, the labels were not used to catch label errors.

The numbers for the axes mappings are very encouraging. In only 22 tables or 3% of the total spreadsheets, the headers were not able to be mapped to axes. This defect was due one of two facts. Either labels were only present on one axis, or the header inference was not able to be determine proper headers for that sheet.

This number is encouraging as it means that most of the sheets will be able to be used by the system. In general 128 tables mapped to two dimension axes, which means that 22% of the tables with dimensions will not have different results when the integrated system is run on them. This is another encouraging number since it means that a vast majority of sheets will be able to gain more error checking capabilities with the integrated system.

## 6. Related Work

The system that is most closely related to out work is SLATE [6], which separates the unit from the object of measurement and defines semantics for spreadsheets so that the unit *and* the object of measurement are considered.

SLATE is the only systems that attempts to measure both labels and dimensions, and it does this by assigning three attributes to every expression: a value, a unit, and a label. The value is what is contained in a cell. Units, such as meters, kilograms, and seconds, capture information about the scale at which the measurement was taken and the dimensions of the measurement. The final attribute, labels, defines characteristics of the objects of measurement. For example, a cell referring to 25 pounds of apples might read "25 lbs. (apples)".

For this system to work correctly it requires a user to annotate a spreadsheet, which involves adding the units and labels to every non-formula cell. The system then analyzes the formula cells and determines the unit and label for these cells. This information is then displayed in the cell. One of the primary problems with this approach is that is does not actually detect errors, it simply displays labels and units for each cell.

Another related system is XeLda [4] which is designed to check a spreadsheet for units of measurement, such as meters, grams, and seconds. Much like SLATE, XeLda requires the user to annotate the units for all of the cells in a spreadsheet. Note that this does not only include data cells, but also all formula cells. While analyzing a spreadsheet XeLda checks the annotated units against the results of formulas to insure correctness.

The advantage of the XeLda approach is that it works well independently of the spreadsheet layout, whereas our approach depends on header and label analysis. On the other hand, XeLda's disadvantage is the huge amount of extra work required by the user whereas our approach is fully automatic. Moreover, XeLda cannot infer conversion factors.

With respect to the error discussed in the Section 2 that requires a combined label and dimension analysis for its discovery, SLATE would infer a proper dimension together with a label (VW Bug, Camry), which is meant to draw the attention of the user and hopefully point out something that might be wrong with the formula, but it is not marked as an error. XeLda would not be able to catch this error as it depends entirely on units of measurement annotated by the user.

UCheck [2] was designed to check for labels in a spreadsheet, and as such it does not handle dimensions. UCheck works by inferring headers for all the cells in a spreadsheet, based on the structure and content of the spreadsheet. Once

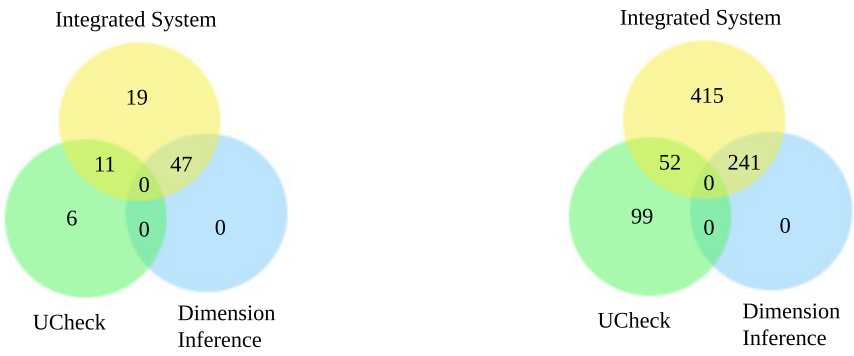


**Figure 3. System comparison: Identified erroneous spreadsheets (left) and total errors (right)**

these headers are inferred, the system derives labels for the cells and checks for label errors.

While UCheck works completely automatically, some other related approaches require the user to annotate the spreadsheet with label information [7, 3]. The same advantages and disadvantages that we have mentioned for XeLda apply here as well.

## 7. Conclusions and Future Work

We have introduced a system that integrates label-based reasoning with dimension analysis. This integration has strengthened the system considerably with the evaluation showing almost a twofold increase in the number of errors that the system is able to detect.

In future work, we will try to strengthen the integration of UCheck and dimension inference further. We have seen that some label errors were missed because some of the labels were exclusively used for dimension analysis. We could potentially catch more errors by running the combined system *and* UCheck and taking the union of errors reported by both systems. However, this might also increase the chance of hitting more false positives. To study different options for an integrated system, we can employ the system architecture that we have developed in previous work [9].

## References

[1] R. Abraham and M. Erwig. Header and Unit Inference for Spreadsheets Through Spatial Analyses. In *IEEE Int. Symp. on Visual Languages and Human-Centric Computing*, pages 165–172, 2004.

[2] R. Abraham and M. Erwig. UCheck: A Spreadsheet Unit Checker for End Users. *Journal of Visual Languages and Computing*, 18(1):71–95, 2007.

[3] Y. Ahmad, T. Antoniu, S. Goldwater, and S. Krishnamurthi. A type system for statically detecting spreadsheet errors. *Automated Software Engineering, International Conference on*, 0:174, 2003.

[4] T. Antoniu, P. A. Steckler, S. Krishnamurthi, E. Neuwirth, and M. Felleisen. Validating the unit correctness of spreadsheet programs. In *ICSE '04: Proceedings of the 26th International Conference on Software Engineering*, pages 439–448, Washington, DC, USA, 2004. IEEE Computer Society.

[5] C. Chambers and M. Erwig. Dimension Inference in Spreadsheets. In *IEEE Int. Symp. on Visual Languages and Human-Centric Computing*, pages 123–130, 2008.

[6] M. J. Coblenz, A. J. Ko, and B. A. Myers. Using objects of measurement to detect spreadsheet errors. In *VLHCC '05: Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 314–316, Washington, DC, USA, 2005. IEEE Computer Society.

[7] M. Erwig and M. M. Burnett. Adding Apples and Oranges. In *4th Int. Symp. on Practical Aspects of Declarative Languages*, LNCS 2257, pages 173–191, 2002.

[8] M. Fisher and G. Rothermel. The euses spreadsheet corpus: a shared resource for supporting experimentation with spreadsheet dependability mechanisms. In *WEUSE I: Proceedings of the first workshop on End-user software engineering*, pages 1–5, New York, NY, USA, 2005. ACM.

[9] J. Lawrence, R. Abraham, M. M. Burnett, and M. Erwig. Sharing Reasoning about Faults in Spreadsheets: An Empirical Study. In *IEEE Int. Symp. on Visual Languages and Human-Centric Computing*, pages 35–42, 2006.

[10] K. Rajalingham, D. Chadwick, B. Knight, and D. Edwards. Quality control in spreadsheets: A software engineering-based approach to spreadsheet development. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 4*, page 4006, Washington, DC, USA, 2000. IEEE Computer Society.

[11] C. Scaffidi, M. Shaw, and B. Myers. Estimating the numbers of end users and end user programmers. In *VLHCC '05: Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 207–214, Washington, DC, USA, 2005. IEEE Computer Society.