

Reasoning about Spreadsheets with Labels and Dimensions*

Chris Chambers
Oregon State University
chambech@eecs.oregonstate.edu

Martin Erwig
Oregon State University
erwig@eecs.oregonstate.edu

July 10, 2010

Abstract

Labels in spreadsheets can be exploited for finding formula errors in two principally different ways. First, the spatial relationships between labels and other cells express simple constraints on the cells usage in formulas. Second, labels can be interpreted as units of measurements to provide semantic information about the data being combined in formulas, which results in different kinds of constraints.

In this paper we demonstrate how both approaches can be combined into an integrated analysis, which is able to find significantly more errors in spreadsheets than each of the individual approaches. In particular, the integrated system is able to detect errors that cannot be found by either of the individual approaches alone, which shows that the integrated system provides an added value beyond the mere combination of its parts. We also compare the effectiveness of this combined approach with several other conceivable combinations of the involved components and identify a system that seems most effective to find spreadsheet formula errors based on label and unit-of-measurement information.

Key Words: Spreadsheet, Dimension, Unit of Measurement, Static Analysis, Inference Rule, Error Detection

1 Introduction

Spreadsheets are widely used [1] end-user programs that contain many errors [2]. To improve the quality of spreadsheets a variety of approaches to prevent, detect, and remove errors from spreadsheets have been investigated. Since preventive approaches, in principle, have to interfere with the creation process that makes spreadsheets so attractive to end users, much research has instead focused on the detection and removal of errors.

One type of error that can be detected in spreadsheets is dimension errors, which occur when units of

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

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 [3], 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 [4, 5, 3] have been developed in order to deal with dimension errors. Among these dimension inference [4] 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 [6] or the system described in [7], 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 in formulas by identifying inconsistent labels.

One thing to note when looking at UCheck and dimension inference is that both systems rely on header and label information. However, this information is used quite differently by the two systems. 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 [5]. 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 together with the additional time required of a user to annotate a spreadsheet renders the approach currently impractical.

In recent work [8] we have designed an integrated system that combines label-based reasoning with dimension inference. This approach was achieved by gathering both label and dimension information about a cell. In many cases a spreadsheet contains dimensions on only one axis, with the labels on the dimension free axis going unused in dimension inference. These unused labels can help to provide structural information that can be exploited by the reasoning system behind UCheck, and create a system that can check a spreadsheet for unit of measurement errors as well as detect label errors.

When run on the EUSES repository this combined system was able to detect several errors that neither UCheck nor dimension inference were able to discover. This brief evaluation shows the validity of a combined label and dimension checking system, but was this the best possible combination?

In previous work [9], a system architecture was developed to study how two specific systems could

be integrated, in this case, WYSIWYT and UCheck. In this work, errors are introduced into spreadsheets and several combinations of the systems are tested to determine the best possible combination of reasoning systems. The process showed that when systems are combined the results are positive and more errors can be detected.

In this paper we will describe several different options, along with the original integrated label and dimension analysis [8], that can be pursued to check a spreadsheet for label and dimension errors. We will evaluate these different systems by running them on the 487 dimension and formula contained sheets in the EUSES corpus, as determined in [4] and determine the efficiency and correctness for each combination.

The rest of this paper is structured as follows. In Section 2 we illustrate the issues involved in adding label reasoning to dimension inference with a small example. In Section 3 we formalize spreadsheets and present models of dimensions and labels. The original combined analysis method is described in Section 4. In Section 5 we describe the different modifications made to the system. Section 6 reports on the evaluation of these prototypical implementations. We discuss related work in Section 7 and give conclusions in Section 8.

2 A Motivating 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 based on the result of a drive using only five gallons of gas.

	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

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 $E2 \cdot D3$. 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 does not 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). Therefore, 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, as shown in Figure 2. The cells F4 and F5 are shaded yellow to indicate problems caused by the propagation of errors.

	A	B	C	D	E	F	G
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	5	=E4*D4	
5		=SUM(B2:B4)				=MAX(F2:F4)	
6							
7							
8							
9							
10							
11							

Error:
Miles and Gallons can not be added

Error:
The labels in this formula do not match

Figure 2: Results for Example Spreadsheet

3 Representations for Spreadsheet Analysis

In this section, we will formalize the notions of spreadsheets, dimensions, and labels in preparation for the formal rule system that is discussed in Section 4.4.

3.1 Spreadsheet Syntax

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.

$$e ::= v \mid \uparrow a \mid e + e \mid e * e \mid \mathbf{count}(e, \dots, e) \mid \mathbf{if}(e, e, e)$$

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 (δ). The *identity dimension* $\{\}$ is used for dimensionless values.

$$\begin{aligned} d &::= \{c, \dots, c\} \\ c &::= b_f^n \mid \delta \end{aligned}$$

For each base dimension we identify a default unit with factor 1. For example, the default for length is meter (m), that is, $m = \text{length}_1^1$, which also means that $\text{cm} = \text{length}_{0.01}^1$ and $\text{ft} = \text{length}_{0.3}^1$. 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 base units from the International System of Units (SI) [10], the modern metric system of measurement used throughout the world. To discover any dimensions that were not covered by these seven base units, a script was run over the EUSES corpus [11] which identified two new base dimensions, Money and Angle. Combined these nine base dimensions were able to cover all dimensions found in the EUSES corpus.

Examples of composite dimensions are speed, measured in m/s, that is $\{\text{length}, \text{time}^{-1}\}$, or force, measured in kg m/s^2 , which is $\{\text{mass}, \text{length}, \text{time}^{-2}\}$.

The relationship between basic and derived dimensions and units is illustrated with several examples in Table 2. Default units are shown as bold.

While this approach of representing conversions by a simple factor is not general enough to cover some conversions, specifically degrees Fahrenheit to degrees Celsius, it keeps the unification of dimensions feasible and works in most cases. Previous work [4] has shown that this restriction is not too severe as only 2 out of 487 spreadsheets contained dimensions that could not be converted using the model presented above.

Table 1: Base dimensions with default units

Quantity	Default Unit
length	meter (m)
mass	kilogram (kg)
time	second (s)
electric current	ampere (A)
temperature	kelvin (K)
amount of substance	mole (mol)
luminous intensity	candela (cd)
money	dollar (\$)
angle	degree (deg)

Table 2: Basic and derived dimensions and corresponding units

	Dimension	Units
basic	length	m , cm, km, ft, ...
	time	s , min, hr, ...
	mass	kg , pounds, ...

derived	speed	$\frac{m}{s}$, $\frac{km}{hr}$, ...
	force	$\frac{kg\ m}{s^2}$ = Newton, dyne, ...
	pressure	$\frac{kg}{m\ s^2}$ = Pascal, psi, atm, ...

3.3 Labels

The labeling structure in this integrated system is a simplified version of the formal model presented in [12]. 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 $\ell_1 | \ell_2$ and the vertical bar $|$ to separate grammar alternatives.)

$$\ell ::= v | \ell_1[\ell_2] | \ell_1 | \ell_2 | 1$$

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|Camry|BMW]”. Since OR distributes over label chains [12], we can factor this expression to “Car[VW Bug|Camry|BMW]”. This label expression can be generalized to “Car” [6].

The ADD rule requires a compatibility of labels expressed by the following label simplification rule, $\ell \wr \ell \rightarrow \ell$, defined as follows.

$$\begin{aligned} \ell \wr \ell &\rightarrow \ell \\ \ell \wr \ell[l_1] &\rightarrow \ell \\ \ell[l_1] \wr \ell &\rightarrow \ell \end{aligned}$$

This operation works only for specific arguments, and if it fails in the premise of an inference rule, then a label error has been identified.

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 use labels to determine errors, they use this information in quite different ways.

UCheck essentially exploits the relative position of labels, but it does not 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 C1 is renamed to “Hours”, the formulas in column D will not be dimension correct as the label for that column is “Miles / Gallon”, but the resulting dimension is “Miles / Hours”. To keep the formula dimension correct the labels in D1 and E1 would have to be renamed as well.

The simple act of changing these three labels, to the sheet shown in Figure 3, now changes the entire meaning of the spreadsheet. Despite the fact that no values were changed the data now represents something entirely different, as the the spreadsheet is now calculating how many miles a car can drive based on the speed measured over a shorter sample distance. This simple example shows how dimension inference uses the meanings of labels to gather information and how a label change can make a large impact on the meaning

of a spreadsheet.

	A	B	C	D	E	F
1	Car	Miles	Hours	MPH	Total Hours	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 3: An Example Demonstrating the Meaning of Labels

4 Integrated Label and Dimension Analysis

Integrated label and dimension analysis (or LD analysis for short), while strongly based on both UCheck and dimension inference, has been designed as an add-on to dimension inference. As an add-on, LD analysis goes through much the same process as dimension inference, with a few changes. The five distinct steps that the LD analysis performs while checking a spreadsheet are shown below and discussed in greater detail in this section.

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

Header inference, label interpretation, and dimension instantiation are components that we have adopted unchanged from our previous work [4], and they are therefore only briefly described here. Steps 3 and 4 will be described in greater detail in the following. The last step applies only in those cases when the step 4 produces underspecified dimensions, that is, when it results in inferred dimensions that contain dimension variables.

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.

Figure 4 shows the identified headers, outlined in blue, for the example in Figure 1. Looking at specific cells, B1 is a header for B2, B3, and B4, that is, $H^{-1}(B1) = \{B2, B3, B4\}$, and A2 and B1 are headers of B2, that is, $H(B2) = \{A2, B1\}$. These specific relationships are shown with the arrows in Figure 4.

	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 4: Example Headers

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 [13, 6]. In the context of this paper we simply reuse those techniques.

4.2 Label Interpretation

In the second phase of LD analysis 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 interpretation. 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 “Gallon”. 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 δ , 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 defining dimensions and the other defining 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 6.

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 interpretation allows us to 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. To assist in this explanation Figure 5, which shows a simple spreadsheet containing three tables, will be used. Each of the three tables is highlighted and illustrates an example of one of the three potential configurations of the axes.

	A	B	C	D	E	F	G	H	I
1	Product	Availability		Product	\$		Time Period	Lumber Sold	Total \$
2	Lumber	10		Lumber	50		Day	5	=H2*\$E\$2
3	Tree	3		Tree	400		Week	25	=H3*\$E\$2
4	Concrete	100		Concrete	10		Month	79	=H4*\$E\$2
5									

Figure 5: Axis Examples

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.

An example of this is shown in the first table in Figure 5. This table contains no units of measurements in any of the axes, and if dimension analysis were run, it would garner no results.

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 a simple example, the second table in Figure 5 proves instructive. In this case the unit of measurement, the highlighted cell E1, is contained in the horizontal axis, with the vertical axis being devoid of any dimensions. In this case, we would assign dimensions from horizontal axis, row 1, and labels from the vertical axis, column D.

The majority of spreadsheets and tables investigated have this configuration, with 439 out of the 567 tables containing an axis setup that falls into this category. For a larger 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 [13]). 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 in [4].

The final table in Figure 5 shows a table that contains units of measurement in both axes. In this case, the horizontal axis, row 1, contains the dimension, “\$”, in cell I1, and the vertical axis contains three cells with a specific time period, such as day. In spreadsheets such as this the integrated analysis will run dimension analysis and ignore labels.

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 6. In dimension inference an error would be detected if and only if the dimensions in a formula did not meet the specifications of the rule being applied. 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 together 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 .

$\boxed{S, L \vdash a : \ell}$ <p>LABHDR</p> $\frac{L(a) = \{a_1\} \quad S(a_1) = \ell}{S, L \vdash a : \ell}$	$\boxed{S, D \vdash a : d}$ <p>DIMHDR</p> $\frac{D(a) = \{a_1\} \quad S(a_1) \Rightarrow d}{S, D \vdash a : d}$		
<p>NOHDR</p> $\frac{H(a) = \emptyset}{S, H \vdash a : 1, \delta}$	$\boxed{S, H \vdash a : \ell, d}$ <p>COMBOHDR</p> $\frac{S, L \vdash a : \ell \quad S, D \vdash a : d}{S, H \vdash a : \ell, d}$		
$\boxed{S, H \vdash e : \ell, d}$			
<p>VAL</p> $\frac{}{S, H \vdash v : 1, \delta}$	<p>REF</p> $\frac{S, H \vdash (a, S(a)) : \ell, d}{S, H \vdash \uparrow a : \ell, d}$	<p>COUNT</p> $\frac{S, H \vdash e_i : \ell, d}{S, H \vdash \mathbf{count}(e_1, \dots, e_n) : \ell, \{\}} $	<p>IF</p> $\frac{S, H \vdash e_2 : \ell, d \quad S, H \vdash e_3 : \ell, d}{S, H \vdash \mathbf{if}(e_1, e_2, e_3) : \ell, d}$
<p>ADD</p> $\frac{S, H \vdash e_1 : \ell_1, \{b_{f_1}^n\} \cup d \quad S, H \vdash e_2 : \ell_2, \{b_{f_2}^n\} \cup d \quad \ell_1 \wr \ell_2 \rightarrow \ell \quad c_1 = f_1/f \quad c_2 = f_2/f}{S, H \vdash c_1 * e_1 + c_2 * e_2 : \ell, \{b_f^n\} \cup d}$			
<p>MULT</p> $\frac{S, H \vdash e_1 : \ell, d_1 \quad S, H \vdash e_2 : \ell, d_2 \quad d = d_1 \bowtie d_2 \quad \mathcal{V}(d)}{S, H \vdash e_1 * e_2 : \ell, d}$			
$\boxed{S, H \vdash (a, e) : \ell, d}$ <p>CELL</p> $\frac{S, H \vdash e : \ell, d \quad S, H \vdash a : \ell, d}{S, H \vdash (a, e) : \ell, d}$			

Figure 6: Combined System rules

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, $S, L \vdash a : \ell$ says that in the spreadsheet S and given label header L , the location given by address a has label the ℓ . Similarly, $S, D \vdash a : d$ says that in the spreadsheet S and given dimension header D , the location given by address a has the dimension d .

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 ℓ 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 ℓ and dimension d under the given header relationship H .

Due to one axis containing only one unit labels, many of the label rules are much simpler than the original UCheck system. Instead of having to determine if the resulting label is well formed, we can simply check if the labels are the same. If they are, then no error can be detected, if they are different, then an error is reported.

This is particularly useful for multiplication formulas. UCheck was unable to properly handle multiplication and required that all but one label involved had to be the one unit. By integrating labels with dimension inference, this is loosened, and the only restrictions are that all the labels are the same and that the resulting dimension is valid.

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

The first erroneous cell is D4. This cell contains an addition formula and thus will be checked using the ADD rule, shown below.

$$\frac{\text{ADD} \quad S, H \vdash e_1 : \ell_1, \{b_{f_1}^n\} \cup d \quad S, H \vdash e_2 : \ell_2, \{b_{f_2}^n\} \cup d \quad \ell_1 \wr \ell_2 \rightarrow \ell \quad c_1 = f_1/f \quad c_2 = f_2/f}{S, H \vdash c_1 * e_1 + c_2 * e_2 : \ell, \{b_f^n\} \cup d}$$

When this rule is applied to D4 we get:

$$\frac{S, H \vdash B4 : \text{Car}[\text{BMW}], \{\text{Miles}\} \cup \{\} \quad S, H \vdash C4 : \text{Car}[\text{BMW}], \{\text{Gallons}\} \cup \{\} \quad \text{Car}[\text{BMW}] \wr \text{Car}[\text{BMW}] \rightarrow \text{Car}[\text{BMW}] \quad c_1 = f_1/f \quad c_2 = f_2/f}{S, H \vdash B4 + C4 : \text{BMW}, ?}$$

Since the dimensions in this formula are not compatible, namely “Miles” and “Gallons” can not be added, a dimension cannot be derived, and an error is reported.

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

$$\frac{\text{MULT} \quad S, H \vdash e_1 : \ell, d_1 \quad S, H \vdash e_2 : \ell, d_2 \quad d = d_1 \bowtie d_2 \quad \mathcal{V}(d)}{S, H \vdash e_1 * e_2 : \ell, d}$$

When this cell is being checked the rule would be applied with the following:

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

The application of this rule shows that the dimensions in this multiplication are acceptable. When “Gallons” and “Miles / Gallon” are multiplied, the resulting dimension is “Miles”, which is a valid dimension and will not cause an error. However, the labels “Car[VW Bug]” and “Car[Camry]” are not compatible, or in this case they do not match. 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. Variables occur whenever the spreadsheet does not provide enough information to precisely narrow down the dimensions. In these cases we have to find substitutions for the variables to obtain proper dimensions.

In fact, a dimension involving variables describes a whole class of possible dimensions. For example, length_ϕ can be m, cm, or any other length dimension that can be obtained by substituting values for ϕ . Similarly, the dimension $\{m, \delta\}$ can be instantiated to velocity or acceleration using the substitution $\{\delta \mapsto s^{-1}\}$ or $\{\delta \mapsto s^{-2}\}$, respectively.

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. From those valid dimensions we can then select the one that is most common (as indicated by the numbers to be reported in Section 6).

5 Shared Reasoning

While the method described in the previous section has been shown to be effective at detecting errors in spreadsheets [8], it is but one way to combine these labels and dimensions. In fact, this system was designed only as an add on to the dimension inference system, and so in some cases it ignores the search for label errors to focus instead on identifying dimension errors.

This approach is but one method of many that are available to use. It stands to reason that these other methods may have similar success detecting errors and should be explored. To test these possibilities we have modified the original integrated system, LD analysis, in several ways, each of which is described in the following sections.

The options described in this sections are a combination of three different methods for checking spreadsheets for errors. The first method is UCheck, which is designed to look for label errors, the second is dimension inference, which will check a spreadsheet for unit of measurement errors, and the third is LD analysis, which attempts to check for both label and unit of measurement errors.

To conceptualize how these options differ, it is instructive to think of the axes in a spreadsheet as a reasoning resource. This resource can be used only in specific ways. For example, in LD analysis, we assign axes to dimension inference, if they contain units of measurement, or UCheck if they do not.

There are two specific aspects that we can investigate when discussing axes as a reasoning resource.

1. Which axis is made available to which tool

In LD analysis any axis that had a unit of measurement was assigned to dimension inference and any axis without one was given to UCheck. However, a tool could be designed where UCheck and dimension inference are both always given one axis.

2. Integrated or independent

How the resources are reasoned about is important, and there are two types of reasoning. The first is an integrated approach that will take the label and dimension information and combine it to potentially find more errors, in effect making an entirely new system. The other type of reasoning is independent, where the axes are used by a given tool without any information about how the other axis or system is performing.

The following section will describe four different systems that each take advantage of these axis resources in different ways.

5.1 UCheck followed by Dimension Inference

The simplest method of combining labels and units of measurement is to simply run UCheck and dimension inference sequentially. That is, first run UCheck, which will determine the label errors in the spreadsheet, and then run dimension inference to find unit errors.

Once the two systems have been run on the spreadsheet and the errors for each have been found, they can be displayed to the user. In this case, there may be some cells with errors from both systems. If this is the case, then both errors are displayed in the cell. An example of this can be seen in Figure 7.

	A	B	C	D	E	F
1	Car	Miles	Gallons	MPG	Total Gallons	Miles Possible
2	VW Bug	180	5	=B2/C2	12	=E2*D2
3	Camry	225	5	=B3/C3	10	=E3*D3
4	BMW	300	5	=B4/C4	15	=E4*D4
5		=B2+C3+B4				=MAX(F2:F4)
6						
7						
8						
9						
10						
11						

Label Error:
The labels in the formula are not compatible

Dimension Error:
Miles and Gallons can not be added

Figure 7: Results of UCheck followed by Dimension Inference

This combination method is very basic, and in fact, the two systems share no reasoning whatsoever. While this method should be able to find all of the label and unit errors in a spreadsheet, it will not be able to combine the structural data that is found in the labels of a spreadsheet to improve the dimension analysis.

We can say that this method allows UCheck and dimension inference access to both axes, however, none of the reasoning or error detection is shared between the systems. This non-integration may cause fewer errors to be detected, but it will, at the very least, give a baseline with which to judge the integrated analysis methods.

5.2 LD Analysis followed by UCheck

In previous work [8] we have seen that some label errors were missed because some of the labels were exclusively used for dimension analysis. As that system was designed to be simply an add on to the dimension inference tool, this was not an immediate concern. However, it would certainly be preferable to be able to detect as many errors as possible.

The primary weakness of integrated analysis is the lack of access to the complete structural information of a spreadsheet, which can cause some label errors to go undetected. An easy way to alleviate this problem is to first run integrated analysis to find label and unit errors. Finally, run UCheck on the spreadsheet.

Running UCheck separately allows the system to detect label errors that may not have been detected by the limited label checking of LD analysis. It should be noted that this may result in more false positives or errors being reported multiple times.

The results from this combination could be displayed in a similar fashion as in the system described in the previous section, with cells containing errors from both LD analysis and UCheck displaying two error messages.

We can say that this method allows LD analysis and UCheck access to both axes. As LD analysis already incorporates dimensions and labels in an integrated reasoning method, the additional label checking does not need to be integrated and can remain independent.

5.3 UCheck Enhanced with Dimension Inference

UCheck Enhanced with Dimension Inference (UCheck⁺ for short) is based on UCheck and uses dimension inference to add some dimension checking. Previous research [8] has shown that the majority of dimensions are found on only one axis. Specifically, only 128 out of the 567 tables in the investigated spreadsheets contained dimensions on more than one axis. It stands to reason that only one axis may be required to detect the majority of dimension errors in spreadsheets.

UCheck⁺ allows UCheck to have access to both axes, and will grant dimension inference access to just one. Formulas will then be checked using a form of integrated reasoning accessing both labels and dimensions.

To explain UCheck⁺, Figure 1 will be used as an example. In this sheet, all of the dimensions can be found on the horizontal axis in row 1. For this particular method, row 1 would become the dimension axis and the cells in the spreadsheet would have dimensions determined based on the headers in this row. In this case the other axis, Column A, does not contain dimensions so this problem is a rather simple one and we can simply give labels to cells in the same manner as UCheck.

Since UCheck has access to both axes, the labels assigned now contain AND labels [12]. For example, in Figure 1 the label for B4 is now “Car[BMW]&Miles”, and the dimension is “Miles”. In the case of Figure 1, UCheck⁺ would be able to detect both of the errors. In the first error cell, D4, the dimensions of the cells being added are still not compatible, and in the second, the inclusion of dimensions allows UCheck to loosen its rules on multiplication, and as the labels don’t match, the error is detected.

In the case that dimensions exist on both axes, the system will choose the axis that affects the highest number of cells. Take for example Figure 8, which shows a spreadsheet containing dimensions on both axes. In the horizontal axis there are 2 cells containing a dimension and in the vertical axis there are 3.

To determine how many cells are affected by each axis we look at the cells each is a header for. For example, the cell A2 is a header for all the cells in row 2, or 3 in total. This is the case for every header in

	A	B	C	D
1	Product	Price	Items Sold	Total \$
2	Apples (1 Pound)	\$1.50	50	\$75.00
3	Oranges (1 Pound)	\$2.25	23	\$51.75
4	Bananas (1 Bunch)	\$2.00	70	\$140.00
5	Cranberries (8 Grams)	\$0.50	15	\$7.50
6				

Figure 8: Axes Example for UCheck⁺

column A, so in total the cells A2, A3, and A5 affect 9 total cells. Looking at the horizontal axis (row 1) we can see that the cells B1 and D1 each affect 4 cells, meaning the entire axis affects 8 total cells. Since the horizontal axis affects more cells in this case, we shall use this as the dimension axis.

5.4 Fully Integrated Analysis

The final modification is to use both axes to determine both the labels and the dimensions. In LD analysis any axis that contained a unit of measurement is used to determine units of measurement in the spreadsheet, and any axis that does not have a unit of measurement is then used to determine the labels in a spreadsheet.

While this method can be successful, it might ignore useful label information in some cases. However, by using both axes to determine the labels and dimensions of a spreadsheet we are able to use all the structural information that the label checking can provide without losing any dimension checking ability.

As an example, we will look at the cell B4 in Figure 1. In fully integrated analysis the two axes are used to assign both labels and dimensions. So this cell will have the label “Car[BMW]&Miles” and the dimensions is “Miles”. Should the spreadsheet have dimensions on both axes, the dimensions would be combined via the method described in [4].

To create this system the rules for integrated analysis have to be modified. Since the entire structural information of the spreadsheet is being used, the rules get a little more complex. For example, in each rule we now have to check to make sure that the label that is created is well formed and that the dimensions are compatible. In addition we still check for errors using integrated reasoning.

6 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 [13] of the UCheck tool [6] and is based on the dimension inference tool presented in [4]. In addition we have created four modified systems, described in Section 5, that attempt to approach the problem in new ways. In this section we describe an evaluation of all of these system to answer the following research questions.

RQ1: *Which system performs best at detecting errors in spreadsheets?*

Each of these systems has strengths and weaknesses that should become evident from the results of our evaluation. Will the results show one as being able to detect more errors than others? Or will there be no clear delineation between the different analysis options?

RQ2: *How do the systems handle false positives?*

With the combination of systems it stands to reason that this may increase the number of false positives (or false negatives) that are reported. Will the systems show the same number of false positives or will there be notable differences?

RQ3: *Are both axes needed to detect unit of measurement errors?*

In one combination we are limiting labels to only one axis. What effect will this have on detecting dimension errors? Will this make this method invalid or will the improvement in detected label errors overshadow the dimension errors that are no longer detected (if there are any)?

6.1 Experiments

To answer these research questions we have employed the EUSES spreadsheet corpus [11], which currently contains 4498 spreadsheets collected from various sources. Of these 4498 spreadsheets only 487 contain both dimensions and formulas. In previous work [8], dimension inference, LD analysis, and UCheck were run on all of these spreadsheets to determine how well each system fared and how many errors each found. In these results only 83 of those 487 spreadsheets contained any errors, so it is to these sheets that we will limit our experiment, as these are the only sheets where any errors were detected by the three systems.

To run our experiment we ran all of the systems on these 83 spreadsheets. To verify the results from the previous evaluation, UCheck, dimension inference, and the original LD analysis were all run again. In addition to these three methods, the four systems described in Section 5 were also run on these spreadsheets to determine how effective they are at detecting errors and how they compare to each other.

For RQ1 the results for each system are compared to determine how many errors were detected and how many each system could not find. These numbers will show how well the systems perform looking at specific errors, such as label errors, as well as overall.

To investigate RQ2, the resulting spreadsheets were checked to determine how many false positives, reported errors that are not actually errors, were identified for each system. To determine this, cells containing errors were looked at and formulas investigated to determine if the reported error is indeed an error.

RQ3 is specifically looking at UCheck⁺, which is the only system that limits dimension inference to one axis. How well this system performs will show whether it is important to use both axes when determining dimensions or if only one is needed.

6.2 Results

The primary task for this evaluation was to run all of these systems on the subset of dimension sheets in the EUSES corpus. This process gave us the necessary information to be able to determine which systems perform better and which errors are not detected by others.

To review and verify the data from previous work we again ran dimension inference, UCheck, and the combined system on these spreadsheets. Dimension inference was able to detect 47 spreadsheets with a total of 241 dimension errors, LD analysis was able to detect 77 spreadsheets with errors and a total of 708 errors, and UCheck identified errors in 17 sheets with 151 total errors. The overlap of errors is shown in Figure 9.

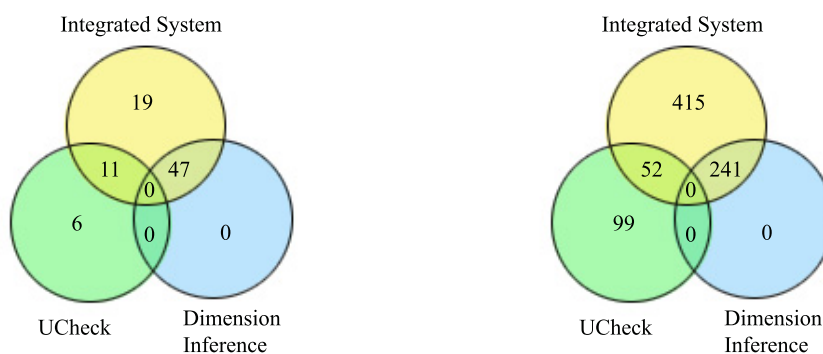


Figure 9: System comparison: Identified erroneous spreadsheets (left) and total errors (right)

To determine false positives, the spreadsheets with errors were looked at more closely. We found that LD analysis 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 LD analysis. Of the 7 false positives, 6 were caused by label interpretation, and one was caused by incorrect header inference. Of the additional 4 false positives in LD analysis one was caused by label interpretation and 3 by incorrect header inference. We did not try to validate the spreadsheets manually, therefore we have no information on false negatives.

In addition to the original three systems, we included several modified systems as described in Section 5. To answer RQ1 and RQ2, the results have been compiled and are summarized in Table 3.

These results reveal the best system to be LD analysis followed by UCheck as it both detects the most errors with a relatively few number of false positives. It should be noted that the false positives are the same number as LD analysis, which demonstrates that this method did not generate any new false positives. The same cannot be said for the fully integrated analysis, which introduced 12 new false positives. However, it

Table 3: System Error Results

Method	Sheets with Errors	Total Errors	False Positives
UCheck	17	151	0
Dimension Inference	47	241	49
LD Analysis	77	708	78
UCheck then Dimension Inference	64	392	49
LD Analysis then UCheck	83	859	78
UCheck ⁺	58	454	25
Fully Integrated Analysis	83	785	90

was able to find the second most number of errors amongst the systems.

The total number of errors and false positives is shown in graphical form in Figure 10. As this chart shows, LD analysis followed by UCheck is able to identify the most errors with a fairly small number of false positives. The first four systems show a fairly significant drop off, which makes some sense considering that either they do not take an integrated approach or they focus on label errors more than dimension errors, which is ill suited for the spreadsheets selected.

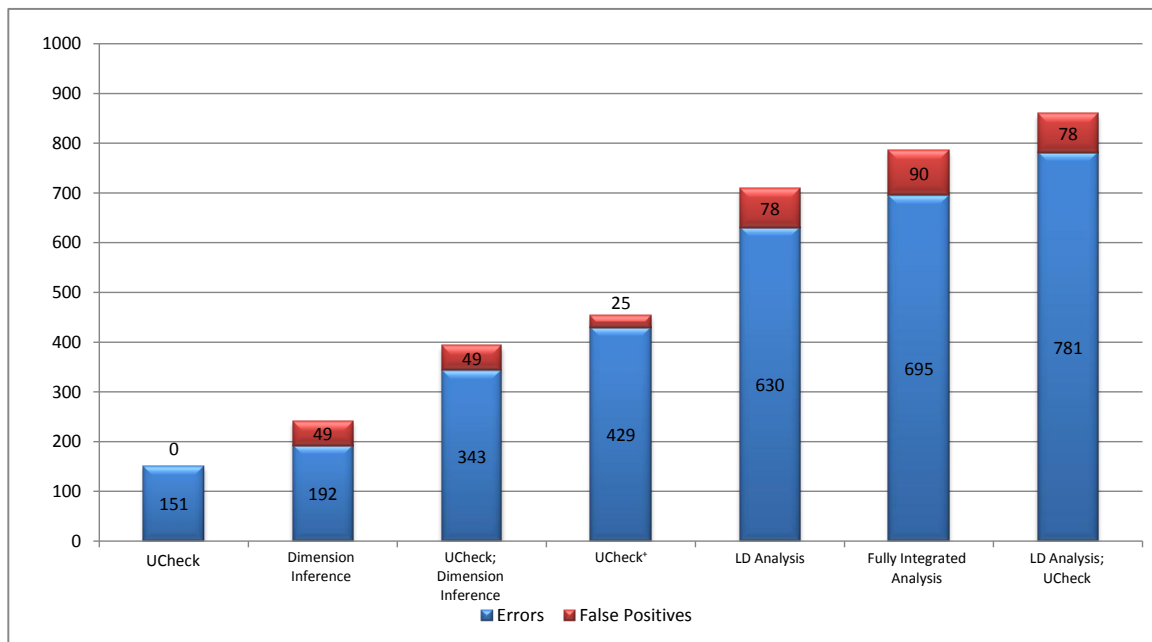


Figure 10: Results of the different Systems

The results of UCheck⁺ show that limiting dimensions to only one axis does make a significant difference in the number of errors that can be detected as there is a significant decrease compared to LD Analysis.

In general we can answer RQ3 and say that two axes are recommended, but probably not necessary to detect dimension errors. However, they are extremely useful.

6.3 Discussion

These results show some interesting trends. Particularly interesting is the strength of the LD Analysis followed by UCheck which was able to detect errors in the largest number of sheets as well as the largest total number of errors. Inspecting the results more closely it can be seen that in a few cases there are label errors that are reported twice. Specifically, there are 52 label errors that were found that were detected by both the LD analysis and by UCheck. While this is not a problem, and could even be considered a feature, it could potentially be confusing to users who may think that there are two errors in the cell.

The other strong system, as was expected, was the fully integrated system, however, it should be noted that this system reports the highest number of false positives and while it is able to detect a high number of errors, this increase in false positives could cause confusion amongst users.

Many of these false positives are due to problems getting the labels to correctly interact with the dimensions. In some cases the system will even miss errors that the combined system caught due to error propagation negating the result.

In general these false positives are caused by two primary problems, an incorrect label interpretation and incorrect header inference. Label interpretation errors occur when a label contains a substring that the tool identifies as a dimension, but it is not. One example of this occurred when a spreadsheet contained a list of names, one of which was "Johnson, Samuel M". The inclusion of "M", caused the tools to interpret this as the dimension "Meters", which caused a false positive to be reported. This type of error is hard to detect, but a combination of manual and automatic methods, discussed in Section 8, may have some success.

The false positives cause by incorrect header inference occur when headers are identified for cells incorrectly. If a header is assigned incorrectly and it contains a dimension it can create problems for the system. There are two possible solutions. The first is the combination of manual and automatic methods that would allow users to change the dimensions assigned by the system. The second is the implementation of header patterns. Identified header patterns could be matched to a spreadsheet and give a better idea of how the headers should be assigned. This could reduce the number of incorrectly assigned headers and false positives.

The approach of simply running UCheck and dimension inference sequentially was rather weak. This system was unable to actually detect any combined errors. It did, however, successfully find all the errors that UCheck and dimension inference found, which is encouraging. Unfortunately, it still lags far behind most of the other systems.

One of the more interesting results was that of UCheck⁺, which limited dimension inference to one

axis. The results from this method indicate that while it was able to detect all of the potential label errors, it was not able to catch many of the dimension or integrated errors. The strength of UCheck⁺ is the full label checking, which may have gone unused in this experiment as there are only 17 sheets containing label errors in the 83 used in this experiment. In general these results show how important two axes are to finding all of the dimension and integrated errors.

It could be noted that previous studies [2] report a much higher failure rate than any of the tools discussed in this paper, even up to 90%. The most effective tool described in this paper reports errors in only 17% of the spreadsheets investigated. This low failure rate is caused by the singular focus of these tools, as they are only looking for two particular kind of errors, label and dimension, instead of focusing on all possible errors in spreadsheets. Often these high error rates are caused by errors such as data entry mistakes which are principally quite hard to detect for any tool.

7 Related Work

The system that is most closely related to our work is SLATE [5], 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 system 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 it does not actually detect errors, it simply displays labels and units for each cell.

The results when SLATE is run on the example spreadsheet used in Section 2 are shown in Figure 11. While the results do not immediately indicate that there is an error, through investigation a user could determine the presence of two. The first is in cell C4, (D4 in Figure 1), has a different dimension than the other cells in the column, (Miles, Gallons) compared to Miles. This is indicative of an error and will hopefully draw the attention of the user. With respect to the second error, SLATE would infer a proper dimension, Miles, together with a label (VW Bug, Camry). This label, which contains the label of two different cars, should illustrate to the user that the cell contains an error. SLATE does not specify what the errors are or

what causes them, but the different labels or dimensions may be enough to make a user curious about the possible reasons and cause a deeper investigation of erroneous cells.

	A	B	C	D	E
1	Miles	Gallons	MPG	Total Gallons	Miles Possible
2	180 (Miles, VW Bug)	5 (Gallons, VW Bug)	=A2/B2 (MPG, VW Bug)	12 (Gallons, VW Bug)	=D2*C3 (Miles, (VW Bug, Camry))
3	225 (Miles, Camry)	5 (Gallons, Camry)	=A3/B3 (MPG, Camry)	10 (Gallons, Camry)	=D3*C3 (Miles, Camry)
4	300 (Miles, BMW)	5 (Gallons, BMW)	=A4+B4 ((Miles, Gallons), BMW)	15 (Gallons, BMW)	=D4*C4 ((Miles, Gallons^2), BMW)
5		=Sum(B2:B5)(Gallons, Car)			=MAX(E2:E4) ((Miles, Gallons^2), Car)
6					

Figure 11: Results of Slate on Example Spreadsheet

Another related system is XeLda [3] 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 interpretation. 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, XeLda would not be able to catch this error as it depends entirely on units of measurement annotated by the user.

While we cannot be certain, it seems likely that a user would annotate F2 with "Miles", E2 with "Gallons", and D3 with "Miles / Gallon". Thus the resulting unit of measurement of the formula would be "Miles", which matches the annotation of the cell, F2. So no error would be reported.

UCheck [6] 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 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 [12, 7]. The same advantages and disadvantages that we have mentioned for XeLda apply here as well.

8 Conclusions and Future Work

In previous work [8] we have introduced a system that integrates label-based reasoning with dimension analysis. This integration strengthened the system considerably with the evaluation showing almost a twofold increase in the number of errors that the system is able to detect. This system was designed to be able to find label and dimension errors, as well as any errors that could be detected with integrated reasoning.

While the individual systems, UCheck [6] and Dimension Inference [4] proved to be useful tools for finding a specific types of errors, the evaluation shown in this paper makes clear that by integrating labels and dimensions a system can detect more errors and can potentially be more useful to the user.

When the results of the individual systems, or even the system where UCheck is run followed by dimension inference, it is clear that without some way to integrate the information there are errors that go undetected. LD analysis worked well as an extension to dimension inference, and as a system it is strong, however it does have the weakness of not using all of the structural information that can be given through labels.

By evaluating several different options, we have identified two that not only integrate labels and dimensions, but also are able to detect the majority of errors. The most effective system is LD analysis followed by UCheck. This system is able to get all of the benefits of integrated reasoning, through LD analysis, as well as the full label checking, through UCheck, that the original system lacked.

The one type of combination that was not explored is that of the manual and automatic approaches. The strength of automatic approaches lay in their ability to run independent of any information provided by the user. However, this can often be inflexible and potentially frustrating to users if the system infers something incorrectly. By incorporating manual approaches and allowing users to provide additional or correctional annotations the system could potentially become and easier to use and stronger application. Such a combination could be explored in future work.

References

- [1] Christopher Scaffidi, Mary Shaw, and Brad Myers. Estimating the Numbers of End Users and End User Programmers. In *IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 207–214. IEEE Computer Society, 2005.
- [2] Kamalasan Rajalingham, David Chadwick, Brian Knight, and Dilwyn Edwards. Quality Control in Spreadsheets: A Software Engineering-Based Approach to Spreadsheet Development. In *33rd Hawaii Int. Conf. on System Sciences-Volume 4*, pages 1–9, 2000.

- [3] Tudor Antoniu, Paul A. Steckler, Shriram Krishnamurthi, Erich Neuwirth, and Matthias Felleisen. Validating the Unit Correctness of Spreadsheet Programs. In *26th IEEE Int. Conf. on Software Engineering*, pages 439–448. IEEE Computer Society, 2004.
- [4] C. Chambers and M. Erwig. Dimension Inference in Spreadsheets. In *IEEE Int. Symp. on Visual Languages and Human-Centric Computing*, pages 123–130, 2008.
- [5] Michael J. Coblenz, Andrew J. Ko, and Brad 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.
- [6] R. Abraham and M. Erwig. UCheck: A Spreadsheet Unit Checker for End Users. *Journal of Visual Languages and Computing*, 18(1):71–95, 2007.
- [7] Yanif Ahmad, Tudor Antoniu, Sharon Goldwater, and Shriram Krishnamurthi. A Type System for Statically Detecting Spreadsheet Errors. *18th IEE Int. Conf. on Automated Software Engineering*, 0:174–183, 2003.
- [8] Chris Chambers and Martin Erwig. Combining Spatial and Semantic Label Analysis. In *VLHCC '09: Proceedings of the 2009 IEEE Symposium on Visual Languages and Human-Centric Computing (VLHCC)*, pages 225–232, Washington, DC, USA, 2009. IEEE Computer Society.
- [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] National Institute of Standards and Technology. The NIST Guide for the Use of the International System of Units, 2010. <http://www.nist.gov/physlab/pubs/sp811/index.cfm>.
- [11] Marc Fisher and Gregg 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.
- [12] 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.
- [13] 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.