A Web-enabled Approach for Generating Data Processors

Jigarkumar Patel, Sohei Okamoto, Sergiu M. Dascalu, Frederick C. Harris, Jr.
Department of Computer Science & Engineering
University of Nevada Reno
Reno, NV, 89557, USA
{jspatel, okamoto, dascalus, fredh}@cse.unr.edu

***Abstract*—Researchers in environmental sciences work with datasets that have a large variety of data structures and file formats. Consequently, for data and model interoperability, it is essential to have the right tools for converting from a given data structure to another, and from a specific file format to another. In this paper, we propose an original web-enabled approach for generating data processors capable of handling a multitude of data operations, including numerous data conversion and processing activities. The proposed approach emphasizes end-user, direct manipulation-based definition of data processors and automated code generation of their associated code, thus freeing the scientists from specialized and often tedious programming tasks. This, in turn, translates into an increased efficiency of the scientific research work. Details of the proposed approach and its supporting web-based software tool are presented in this paper, together with an application example in which an input dataset in the CSV format is converted into an output dataset in the XML format. The applicability of the proposed approach goes beyond the domain of environmental sciences, for which it was initially created, as it can be used in many other areas of research that emphasize data-intensive exploration and processing.**

Keywords-data processors, web-enabled software tool, graphical interface, data and model interoperability.

I. INTRODUCTION

Funded by a couple of recent NSF EPSCoR grants, researchers, educators, and students in Nevada, Idaho and NewMexico have focused on topics pertaining to the effects of climate change on their regional environments and ecosystem resources [1, 2]. In addition to environmental science researchers, cyber-infrastructure (CI) teams (to which the authors of this paper belong) have been substantially involved in these NSF-funded projects.

One of the primary goals of the CI group has been to facilitate and support interdisciplinary climate change research, education, policy, decision-making, and outreach by using CI to develop and make available integrated data repositories and intelligent, user-friendly software solutions [3, 4, 5]. Part of these solutions we have aimed at producing new software tools for data and model interoperability [6, 7, 8, 9], including tools for generating data processors capable of handling a large variety of data conversion and processing activities.

In our prior work [10] we have described the motivation for embarking on creating efficient software tool support for environmental science research, and have outlined the main challenges pertaining to handling geospatial data, namely the diversity of data storage formats, the need for effective data processing (in particular, for filtering, merging, sorting, and grouping), and the complex nature of data scaling on spatiotemporal dimensions.

In the same work we have described the design of an early software solution for data interoperability support; specifically, we have presented the six subsystems of our software system (users management, file formats, data structures, data structure operations, workflows, and dynamic code generation) and discussed the main architectural patterns employed in our solution (model-view-controller and a combination of the multi-tier and the central repository patterns). The principles of the proposed solution's user interaction and several brief details of the tool's user interface were also presented in [10].

Building on our prior work, in this paper we detail the specific steps of the proposed approach and comprehensively present the supporting tool's user interface. Furthermore, to illustrate the usefulness and versatility of the proposed approach, an application that takes as input a dataset in CSV format and produces as output a processed ("filtered") dataset in XML format is also presented. While this example is rather short (because of space limitations), it nevertheless gives an idea of the power of the proposed approach and its supporting web-enabled software tool. In essence, the main contribution of the work presented in this paper is to provide a method and related software for the effective, end-user definition and automated code generation of data processors that support and facilitate scientific exploration, experimentation, and discovery.

Notably, the proposed approach and its associated software tool are not confined solely to the domain of environmental science research but can be effectively used in many other areas of research in which the management of datasets and, in general, data processing activities are particular important (for example, in physics, chemistry, astronomy, neuroscience, and many more).

This paper, in its remaining part, is organized as follows: Section II provides a brief overview of related work, Section III

describes in detail the proposed approach for data processor definition and generation, Section IV presents an application example in which a data processor is created to that convert a dataset in CVS format into a derived dataset in XML format, and Section V finalizes the paper with directions of future work and our conclusions.

II. Related Work

Currently, to access the data they need researchers in environmental sciences have access to a variety of public data warehouses and repositories. For instance, the National Oceanic and Atmospheric Administration [11] and other similar data repositories provide web-based interfaces to download data using text and map-based search tools and selection capabilities. However the scientists do not have control over the output climate variables and do not have many options regarding the format of the data files they wish to download.

Furthermore, for data processing and model simulation purposes, researchers can use sophisticated scientific workflow-based software environments such as Kepler [12, 13]. Kepler allows users to interconnect available computational components and data sources and thus create executable scenarios that support desired research simulations and experimentations. While Kepler is a complex and rich software environment with many excellent capabilities, it may involve a steep learning curve and the workflows it produces require the environment to be running, thus these research workflows (simulation scenarios) are significantly dependent on Kepler's availability and resources.

Several other tools and environments [14, 15, 16, 17] are also available to process locally stored data but they are generally tied to particular data types and usually do not offer straightforward capabilities for extension. The approach we propose in this paper offers some new, straightforward, and versatile ways for dealing with several data interoperability challenges that are not being directly tackled by others.

III. The Proposed Approach

Our work aimed at addressing several challenges in data and model interoperability has resulted in a new web-enabled approach and supporting software for data processor definition and generation. The key characteristics of the proposed approach, partially depicted in Fig. 1, are: (i) a user-friendly, direct manipulation-based, visual definition of new data processors (each data processor consisting of a specific data structure and a specific set of operations on the elements of that data structure); and (b) automatic generation of the data processors' code.

Due to space limitations, only the first three phases of the proposed approach are depicted in Fig.1. They are, namely, *data structure definition*, *data operations definition*, and *data processor creation* (the latter includes data processor definition, code generation, and, optionally, downloading).

In order to describe the approach, the following terms are introduced:

- *Data structure* is a linear collection of data elements/columns (described next). As shown in Figure 2, a data structure has a name (Sensor Data in the case shown in the figure) and, optionally, a description (Data from NV climate portal in the example shown in Figure 2). In general, a data structure defines the organization of a *dataset*.

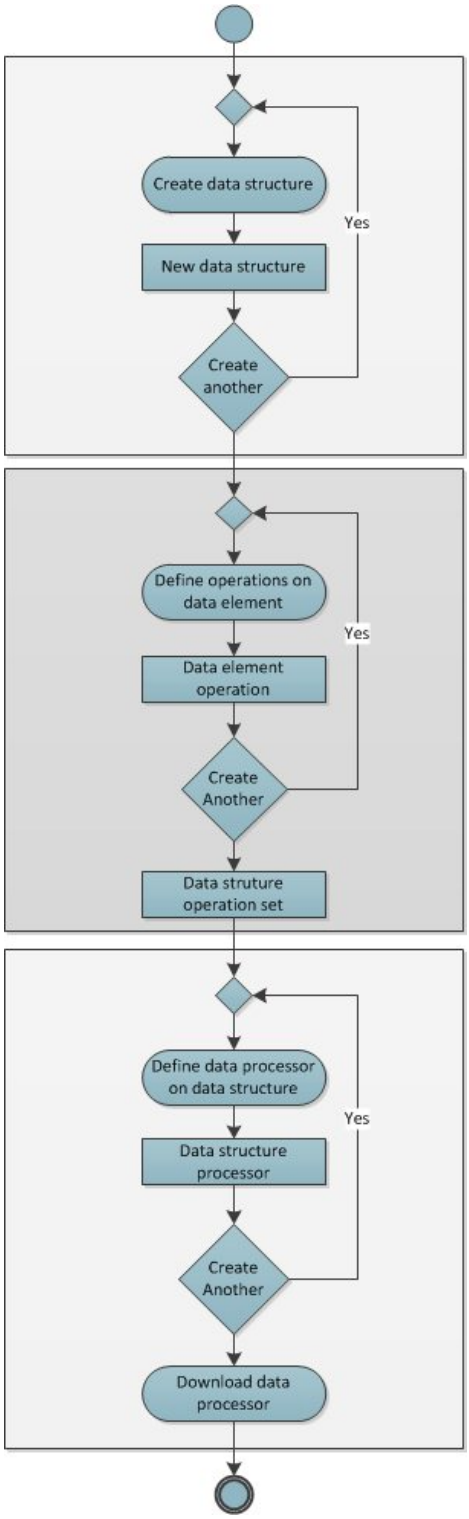

Figure 1. The Overall Approach for Data Processor Definition & Generation

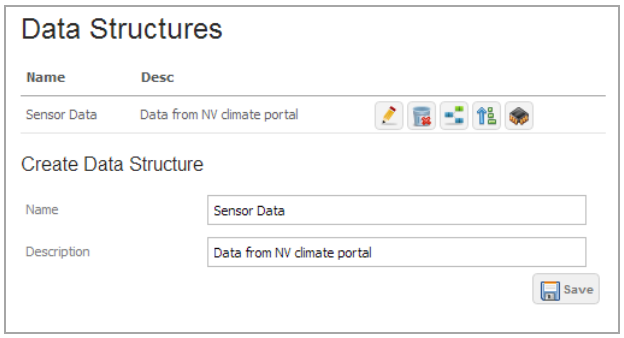

Figure 2. Creating a Data Structure

- *Data element* (or informally called *data column,* because of its role in data tables) is an atomic indivisible) data item in a data structure, for example the variable Station Name in the data structure shown in Fig. 3. Other examples of data elements presented in this figure are Timestamp, Wind Speed, Atmospheric Temperature, and Relative Humidity.

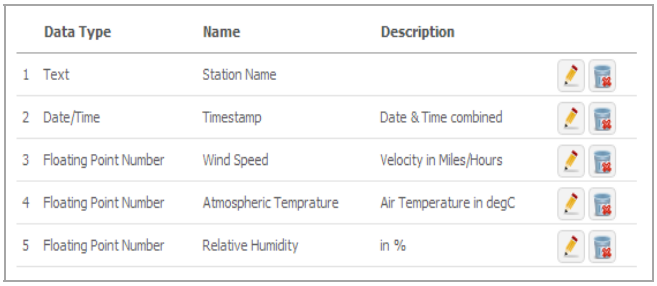

Figure 3. Data Elements/Columns in a Data Structure

- Each data element has associated a specific *data type*, for example in Figure 3 Station Name has the data type Text while Atmospheric Temperature has the data type Floating Point Number. The interface for associating a data tyype to a data elemnet/column is shown in Figure 4. For an end-user's perspective, it suffices to know that at this time available data types in the system are Text, Integer, Floating Point, and variations of Date/Time.

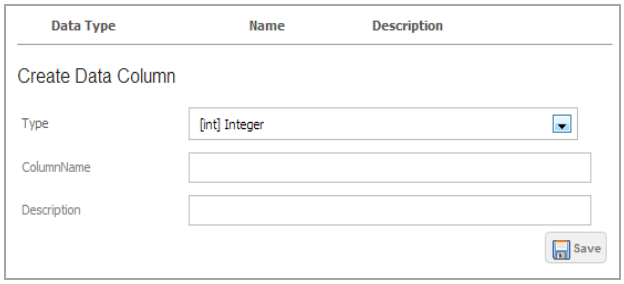

Figure 4. Creating a Data Element/Column

- *Data operation* is a processing operation that can be applied on a data element. Currently, the operations available in the system that we have built fall in the following categories: filter operations (an example is shown in Fig. 5), math functions (an example is presented in Fig. 6), the ignore operation (shown in Fig. 7), and the sort operation.

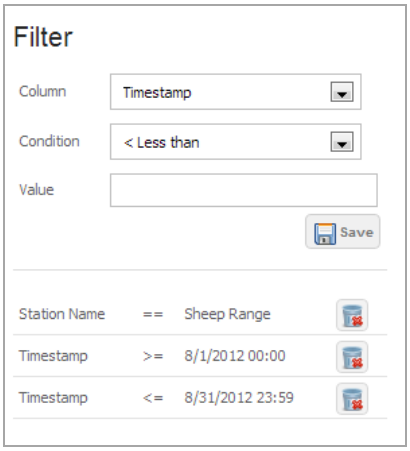

Figure 5. Adding a Filter Operation

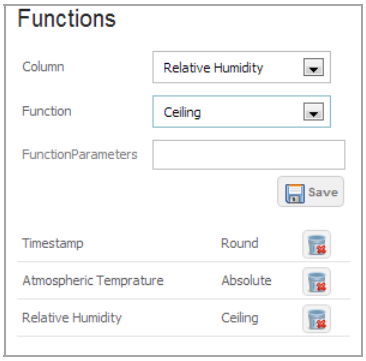

Figure 6. Adding a Math Function Operation

- *Filter operations* are based on a comparison with a specified value, and allow selecting form a data structure (dataset) only those data element values that satisfy the comparision (e.g., Timestamp >= 8/1/2012 00:00 in Fig. 5). *Math functions* that can be applied on data element values include Floor, Ceiling, Round, and Absolute Value. The *Ignore operation* comes very handy when data elements/columns from a given (input) dataset are not needed in the resulting processed (output) dataset. In the example shown in Fig. 7 the Station Name column is not needed in the processed output dataset. Finally, a sort operation (of either Ascending or Descending type) could be applied on a single data element/column of any given data structure.

- Notably, all the operations applied on the data elements of a given data structure constitute an *operation set*, which is assigned a name (for saving, loading, and data processor definition purposes) and optionally can have several comments or a description attached (for documentation purposes).

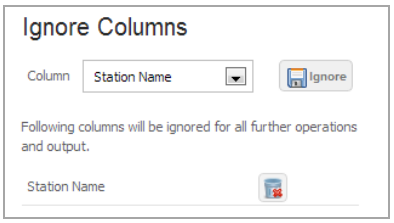

Figure 7. Adding the Ignore Operation

- *Data processor* consists basically of the combination *data structure* + the *operation set* that can be applied on the data elements of the specified data structure. As shown in Fig. 8, for practical purposes the definition of a data processor also includes the *file format* of the Input File (input dataset) and the format of the Output File (output dataset). Currently, the file formats supported by our software, and the underlying code generation capabilities needed for conversions among them, are: CVS, XML, Excel, and ASCII.

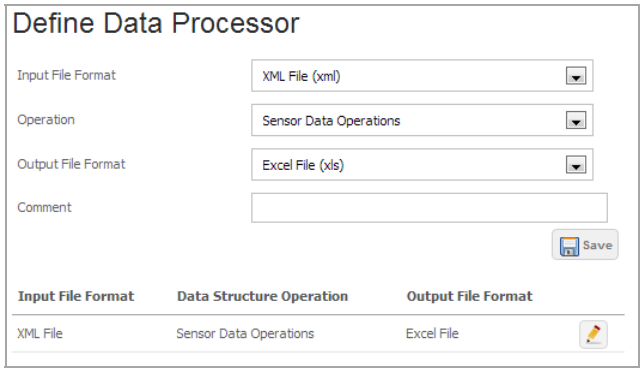

Figure 8. Defining a Data Processor

From a procedural point of view a data processor applies transformations on the input data file (of a specified file format) whose data structure and set of operations have been previously defined. The result of applying these transformations is an output file generated in the format also specified in the definition of the data processor. From the above, it can be seen that the concept of data processor is very powerful and versatile, as it can be adapted to and applied on numerous types on computations, in many scientifci domains that intesively use data and data sets.

IV. Sample Application

To concretely illustrate the proposed approach and its supporting web-enabled software tool, a relatively simple yet representative application is presented next. Note that the previous Figures 2-8 also include details pertaining to this application, and they can be consulted for further details.

In short, the application is the following: given an input dataset in the CSV format and with the structure [StationName, TimeStamp, WindSpeed, AirTemperature, RelativeHumidity], generate a data processor whose application will produce an output file in the XML format, with the structure [TimeStamp, WindSpeed, AirTempetature, and Relative Humidity], and containing data records only for the Sheep Range station and only for measurements taken between 8/1/2012 00:00 and 8/31/2012 23:59. Furthermore, the output dataset should be sorted in ascending order on TimeStamp values and for each record it should show the absolute values of Atmospheric Temperature measurements and the ceiling values of RelativeHumidity measurements.

```
1  StationName, TimeStamp, WindSpeed, AirTemprature, RelativeHumidity
2  Sheep Range, 8/1/2012 01:00, 5.2, 25.5, 59
3  Sheep Range, 8/1/2012 09:00, 2.3, 27.5, 45
4  Sheep Range, 8/1/2012 17:00, 15.4, 32.4, 36
5  Sheep Range, 8/2/2012 01:00, 8.0, 23.7, 61
6  Sheep Range, 8/2/2012 09:00, 4.3, 26.5, 47
7  Sheep Range, 8/2/2012 17:00, 9.6, 33.6, 38
8  Incline Village, 8/1/2012 01:00, 1.5, 18.1, 65
9  Incline Village, 8/1/2012 09:00, 3.2, 20.9, 52
10 Incline Village, 8/1/2012 17:00, 7.8, 24.7, 38
11 Incline Village, 8/2/2012 01:00, 2.3, 19.1, 63
12 Incline Village, 8/2/2012 09:00, 2.1, 21.6, 49
```

Figure 9. Input Dataset File in CSV Format

```
<?xml version="1.0"?>
- <SensorData>
  - <Record>
      <TimeStamp>8/1/2012 01:00</TimeStamp>
      <WindSpeed> 5.2</WindSpeed>
      <AirTemprature> 25</AirTemprature>
      <RelativeHumidity> 59</RelativeHumidity>
    </Record>
  - <Record>
      <TimeStamp>8/1/2012 09:00</TimeStamp>
      <WindSpeed> 2.3</WindSpeed>
      <AirTemprature> 27</AirTemprature>
      <RelativeHumidity> 45</RelativeHumidity>
    </Record>
  - <Record>
      <TimeStamp>8/1/2012 17:00</TimeStamp>
      <WindSpeed> 15.4</WindSpeed>
      <AirTemprature> 32</AirTemprature>
      <RelativeHumidity> 36</RelativeHumidity>
    </Record>
  - <Record>
      <TimeStamp>8/2/2012 01:00</TimeStamp>
      <WindSpeed> 8.0</WindSpeed>
      <AirTemprature> 23</AirTemprature>
      <RelativeHumidity> 61</RelativeHumidity>
    </Record>
  - <Record>
      <TimeStamp>8/2/2012 09:00</TimeStamp>
      <WindSpeed> 4.3</WindSpeed>
      <AirTemprature> 26</AirTemprature>
      <RelativeHumidity> 47</RelativeHumidity>
    </Record>
  - <Record>
      <TimeStamp>8/2/2012 17:00</TimeStamp>
      <WindSpeed> 9.6</WindSpeed>
      <AirTemprature> 33</AirTemprature>
      <RelativeHumidity> 38</RelativeHumidity>
    </Record>
</SensorData>
```

Figure 10. Output Dataset File in XML Format

For this application, the input file looks like the one shown in Fig. 9, in which for simplicity only 12 data records were included (although in general datasets can be several orders of magnitude larger than that). The resulting processed dataset in XML format is shown in Fig. 10.

A visual differential comparison between the given input dataset and the output dataset obtained by applying this particular data processor is presented in Figure 11. Because the CSV format is more compact than XML, it was chosen for representing the differences between the two datasets.

V. Conclusions and Future Work

There are many complex challenges that pertain to data processing in general and to data interoperability in particular [6, 7, 10, 18]. The work described in this paper aims to address some of them by providing extended web-based support for data processing (in view of data interoperability), with particular application to simulation-based research in environmental sciences. With our proposed web-enabled approach and associated software, dynamically generated data processors will allow more effective experimentation and validation of research hypotheses and will provide a foundation for tackling more complex model and data interoperability problems, in a variety of scientific domains.

The proposed approach is based on the concept of data processor, which can be embodied in a multitude of implementations, addressing a large variety of data processing needs, from data structure transformations to data filtering activities, and to file format conversions. The distinguishing characteristics of the proposed approach are, in essence, the user-friendly, direct manipulation-based, visual definition of new data processors and the effective, automatic generation of the new data processors' executable code. The approach and its associated software tool aims to free the scientists from specialized and time consuming programming tasks, thus enabling them to concentrate more on answering various research questions (than on developing tools for performing work to answer such questions).

Notably, the proposed approach can also be applied in domains other than environmental sciences, for which it was initially created, as it can be used in many research endeavors that rely on intensive data-based processing.

Future work that we have already embarked on includes creating new capabilities for interconnecting a variety of data processors and data sources, thus allowing us to get closer to the ultimate goal of our research and development work, that of providing extended support for data and model interoperability for a wider range of research scenarios.

Acknowledgment

This work was made possible through the support provided by the National Science Foundation under Cooperative Agreements No. EPS-0814372 and No. EPS-0919123.

References

[1] Nevada Climate Change Portal (NCCP)/About the Project/Funding. NSF EPSCoR Coop. Agr. No. EPS-0814372 (2012, Nov.) [Online]. Available: http://sensor.nevada.edu/NCCP/The%20Project/Funding.aspx

[2] Collaborative Research: Cyberinfrastructure Developments for the Western Consotrium of Idaho, Nevada, and New Mexico. NSF EPSCoR Cooperative Agreement No. EPS-0919123 (2012, November) [Online]. Available at: http://nsf.gov/awardsearch/advancedSearchResult?PIOrganization=Nevada%20System%20of%20Higher%20Education&

[3] Dascalu, S. "A cyberinfrastructure project: Building the Nevada Climate Change Portal and its SENSOR system," keynote talk at the joint International Conferences SEDE-2012 and ACC-2012, Los Angeles, CA, June 27, 2012.

[4] Dascalu, S. "Cyberinfrastructure developments for climate change science and education in Nevada," invited talk, Research Seminar series at the University of Milano-Bicocca, Italy, January 11, 2012.

[5] Dascalu S. "Imagine a million file cabinets of climate data: The Nevada Climate Change Data Portal," invited talk, the Nevada Climate Change Seminar Series. University of Nevada, Las Vegas, September 7, 2011. [Online]. Available: http://digitalscholarship.unlv.edu/climate_change/6/

[6] Fritzinger, E., Dascalu, S., Ames, D.P., Benedict, K., Gibbs, I., McMahon, M.J., Jr., and F. C. Harris, Jr. (2012). "The Demeter framework for model and data interoperability," *Proceedings of the International Congress on Environmental Modeling and Software (IEMSS-2012)*, Leipzig, Germany.

[7] Okamoto, S., Hoang, R.V., Dascalu, S.M., Harris, F.C., Jr., and N. Belkhatir (2012). "SUNPRISM: An approach and software tools for collaborative climate change research," *Proceedings of the 13th Intl. Conference on Collaboration Technologies & Systems (CTS-2012)*, Denver, CO, pp. 583-590.

[8] Dascalu, S., Fritzinger, E., Okamoto S. and F.C. Harris, Jr. (2011). "Towards a software framework for model interoperability," in *Procs. of the 9th IEEE International Conference on Industrial Informatics (INDIN 2011),* Lisbon, Portugal, IEEE Computer Society, pp. 705-710.

[9] Okamoto S., Fritzinger E., Dascalu S., Harris F.C., Jr., Latifi S., and M. McMahon, Jr. (2010). "Towards an intelligent software tool for enhanced model interoperability in climate change research," *Proceedings of the World Automation Congress (WAC-2010)*, Kobe, Japan, September 2010, IEEE Computer Society, pp. 1/1-6.

[10] Patel, J., Okamoto, S., Dascalu, S.M., and F.C. Harris Jr. (2012). "Web-enabled toolkit for data interoperability support," *Proceedings of the 21th International Conference on Software Engineering and Data Engineering (SEDE-2012),* Los Angeles, CA, pp. 161-166.

[11] National Oceanic and Atmospheric Administration. (2012, November) NOAA Climate Services. [Online]. Available at: www.climate.gov

[12] San Diego Supercomputing Center, "Kepler: an extensible system for design and execution of scientific workflows," in *Proceedings of the 16th International Conference on Scientific and Statistical Database Management*, 2004, pp. 423-424.

[13] The Kepler Project (2012, November) [Online]. Available at: https://kepler-project.org/

[14] D. Kavan and P. Man, "MSTools—Web based application for visualization and presentation of HXMS data," *International Journal of Mass Spectrometry*, vol. 302, no. 1-3, pp. 53-58, 2011.

[15] M. Nilsson, "The DOSY Toolbox: A new tool for processing PFG NMR diffusion data," *Journal of Magnetic Resonance*, vol. 200, no. 2, pp. 296-302, October 2009.

[16] M. Waldhor and E. Appel, "Intersections of remanence small circles: new tools to improve data processing and interpretation in palaeomagnetism," *Geophysical Journal International*, vol. 166, no. 1, pp. 33-45, July 2006.

[17] C. Schwager, U. Wirkner, A. Abdollahi, and P. Huber, "TableButler – A Windows based tool for processing large data tables generated with highthroughput methods," *BMC Bioinformatics*, vol. 10, pp. 235-243.

[18] Okamoto, S. (2011). "SUNPRISM: A software framework for climate change research," *PhD thesis*, Univewrsoity of Nevada, Reno.

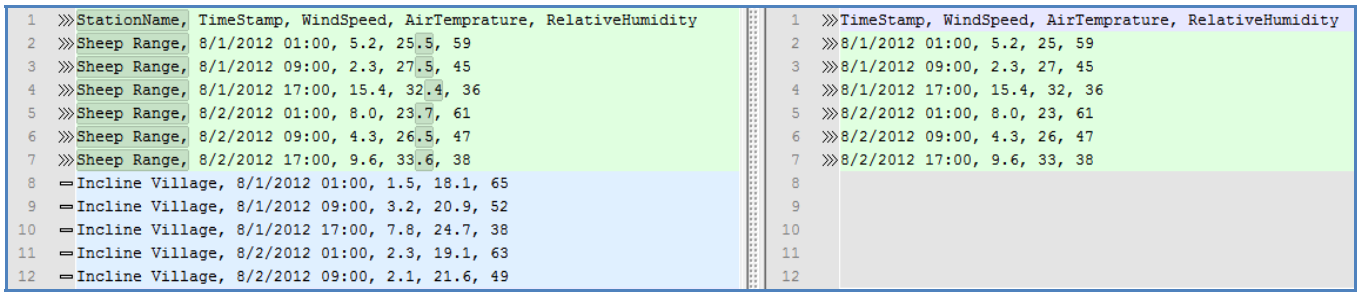

Figure 11. Visual Difference Comparision Between the Input and Output Datasets Presented in the CSV Format