DATA VALIDATION

Ramesh M. Choudhari, South Carolina State University, Orangeburg, SC 29117
Shobha R. Choudhari, South Carolina State University, Orangeburg, SC 29117

ABSTRACT

Data validity is an important attribute of the quality of data. Invalid data can invalidate the findings of the research. The results generated from the data without validating that data can be incorrect and inaccurate. Data validation is very necessary before performing any analysis on the data. So, the researchers developed the system to check data validity based on the inherent and acquired properties of the data. The system can check several things including invalid data type, invalid values based on the type, and out of range values. The system can provide useful information about validation problems related to data objects in a data file. The system will be useful to perform the time consuming and tedious task of validating data, especially for large data files. Thus, the system will be useful for researchers, data managers, and analysts.

INTRODUCTION

Whether it is an educational or business research project, high quality of data is important for producing correct and statistically valid results. Validity is one of the important attributes of quality data. In the context of this paper, data validation is the process of explicitly checking inherent and acquired properties of data. Software using incorrect format or type would result in execution errors causing premature termination of processing. This would cause a significant loss of processing time. Incorrect and incomplete data values would cause incorrect results and subsequently incorrect conclusions. So, complete, consistent, and correct data is essential for research projects. Thus, validity of data becomes the important first step in these projects. In this paper, the researchers would like to focus on validity of the data, and developed the computerized system to check the data validity [2][6][11][15].

CONCEPTUAL FRAMEWORK

Data object can be considered as an abstract object defined by a four tuple (A, N, V, T) where A represents the address of an object in the address space of the computer system, N represents the name of an object in the name space of the software system, V represents the value of an object, and T represents the type of an object. The realization of an object means the determination of all these four entities in the context of the computer system. So, validation can be considered as the process to validate (to check the correctness of) all these four entities. However, data analysis performs analysis on data values. So in the context of data analysis, we will consider only two entities, value and type. Thus, data validation becomes the process of checking the correctness of the value and type of a data object. Type becomes an inherent property of a data object, and user specified properties become acquired properties of a data object.

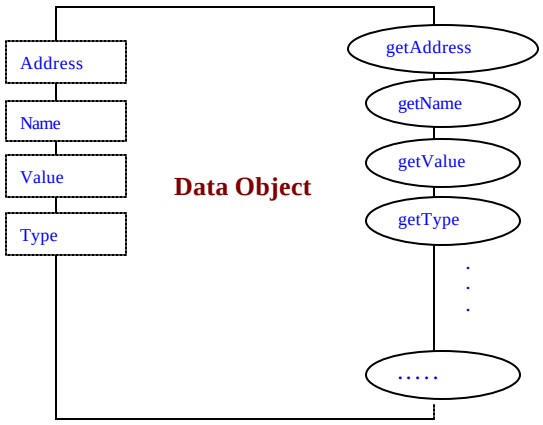

Figure 1. Model of a Data Object

To illustrate the concept, consider the example: If data object is integer type, then a set of integer values (min .. max) and five operations (+, -, *, /, and mod) are its inherent properties. However, if that data object represents age of teenagers in terms of year, then two of its acquired or specified properties could be that its minimum and maximum values are 13 and 17. In the process of validations of such a data object, its inherent property of type integer and acquired property having values in the range of 13 to 17 need to be validated.

Another acquired property of a data object could be the specified length of the values assigned to that data object. The length of the data values can not exceed the specified length, or in some cases data values must have the exact same length, even though higher length is allowed according to the data object's inherent property. A common example of such type is string. We need to check such acquired properties of the data objects.

Format of the data object could also be an acquired property. Many times, format for a data object is specified to enter data. For example, if the data object named "date" has a specific format, say "dd-mm-yyyy", then the data entered for that data object is valid only if it is in the given format even though the other formats of date may represent the same date. Additionally, if the responses for the data object are specified, then the value of the data object is valid only if it is one of the responses specified for that data object. Thus, several required properties can be specified for a data object to validate the data file [7][8][11][15][16][17][18].

DATA VALIDATION SYSTEM

Using the concepts and ideas explained above, the researchers developed the computerized data validation system. The system validates data set by checking inherent and acquired properties of data objects specified by the user. Data set is a collection of data objects. It is realized using text file, referred to as data-file. A data-file is a collection of data-lines. Data-line is a collection of data-fields. Each data-field holds a data object. So the data validation process becomes validation of a data-file. The validation of a data-file is reduced to the validation of data-fields. The model and implementation of the data validation system is explained below.

SYSTEM MODEL

The system model is essentially composed of three modules: User Interface Module, Processing Module, and Output Module.

User Interface Module

The user interface module contains the mouse and keyboard event handlers to collect information from the users regarding the total number of fields in the data-file and specifications about each fields such as field type, length, minimum and maximum values allowed for the data.

Processing Module

The processing module consists of several submodules, each one responsible for processing specific tasks, such as accessing the values of each data-field, processing the user specified properties of the data-fields, and checking if these properties are satisfied by the data objects.

Output Module

The output module is responsible for tasks such as displaying messages along with the data-field and data-line whenever the property of a data-field is not satisfied by its data object, as well as saving the results created by the processing module in an output file [2][3][7][10] [14].

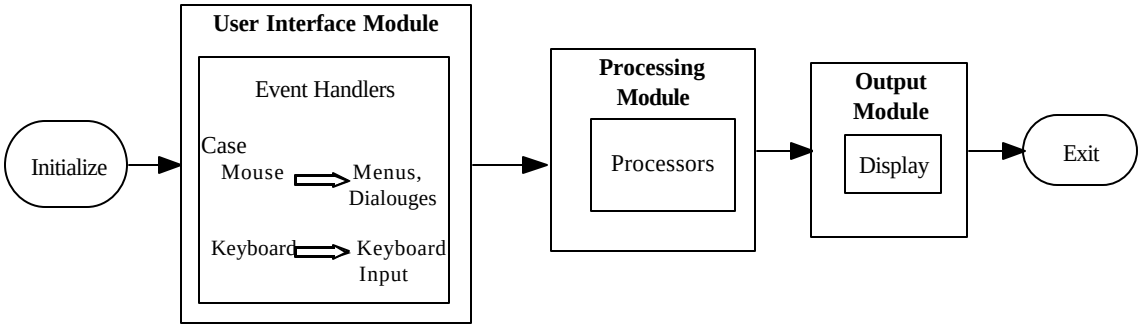

Figure 2. Procedural Architecture of the System Model

IMPLEMENTATION

A prototype of the system is implemented in the Windows environment. The graphic user interface is menu driven, and implemented as menus, using the current GUI techniques. User dialogues are implemented to collect user input as needed.

The heart of the system is the processing module that is implemented by the main processor. The processor consists of several subprocessors that are responsible for specific tasks such as processing user inputs, processing data, and performing necessary computations by applying data validation rules.

Output module is implemented by the display procedures using data aware controls. It displays the appropriate error messages about validation problem along with field and the data-line number in the user friendly format [1][3][4] [5] [9] [12][13][14].

TESTING

The testing of this system was done on more than one data set. The following screen shots show a typical user session with a data file, Main Menu, user dialogues, data-field specifications, and results of data validation. A small data-file with few data-lines was selected for easy understanding of the whole process that covers many categories of data validation.

```
"SSN" "YR"
"1234567899" 2001
"123456789" "X"
"950128843" 2002
"95013063" 2003
"950130889" 2004
"950134851" 2000
"950134851" 2008
"950134851" 2010
```

Figure 3. Screen Shot of "Data-File"

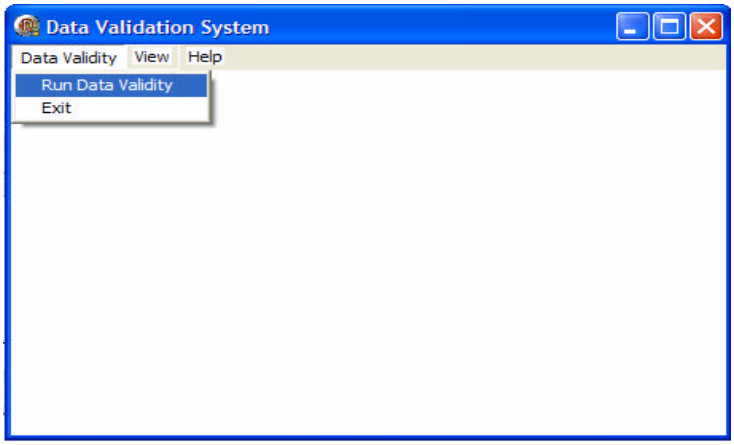

Figure 4. Screen Shot of "Main Menu"

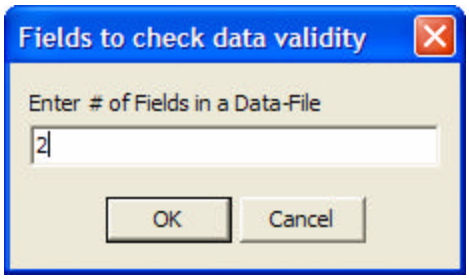

Figure 5. Screen Shot of "User Dialogue 1"

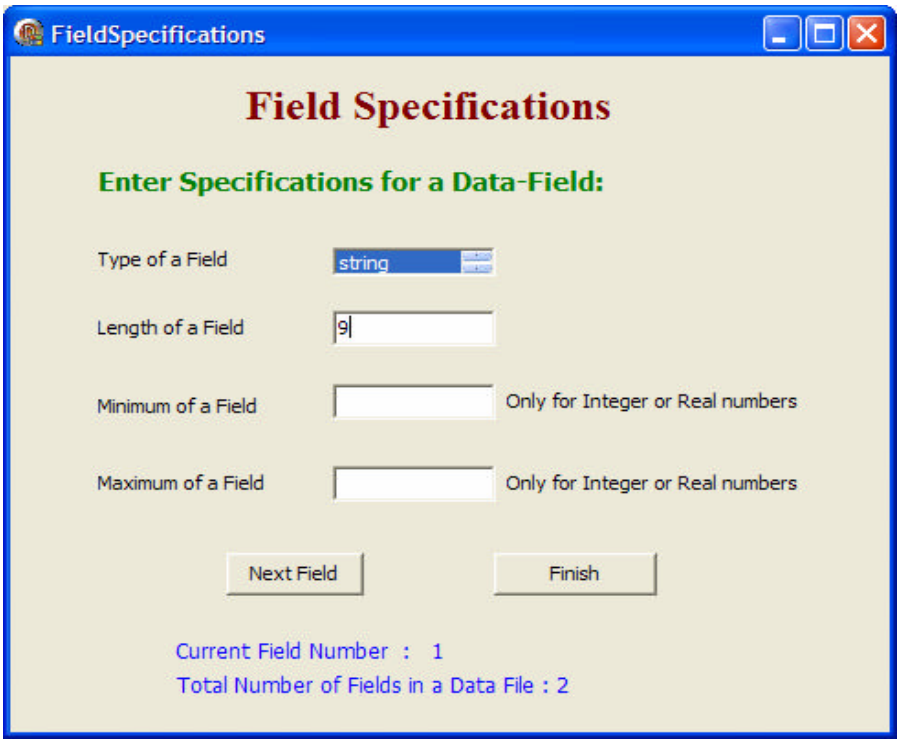

Figure 6. Screen Shot of "Field Specifications"

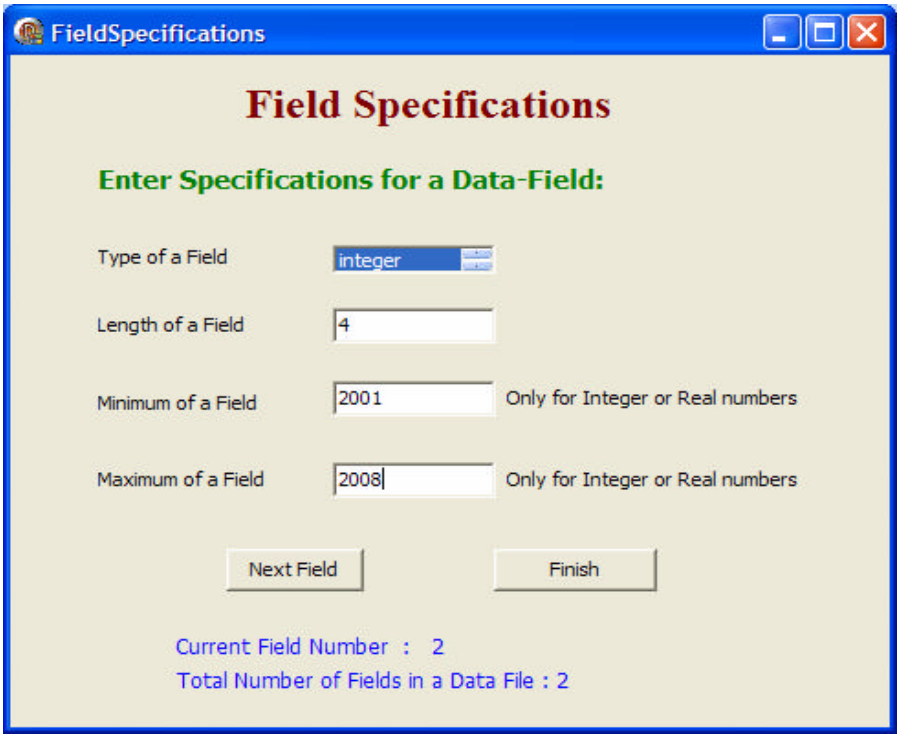

Figure 7. Screen Shot of "Field Specifications"

The error file that reports errors in the data-file after running the data validation system is shown below:

```
"SSN"
    ^ Length Incorrect
"SSN" "YR"
       ^ Type Incorrect
Fields Do Not Match: Error In Line#:       1

"1234567899"
         ^ Length Incorrect
Fields Do Not Match: Error In Line#:       2

"123456789" "X"
           ^ Type Incorrect
Fields Do Not Match: Error In Line#:       3

Correct Line#       4

"95013063"
       ^ Length Incorrect
Fields Do Not Match: Error In Line#:       5
Correct Line#       6

"950134851" 2000
            ^ Value Out Of Range
Fields Do Not Match: Error In Line#:       7

Correct Line#       8

"950134851" 2010
            ^ Value Out Of Range
Fields Do Not Match: Error In Line#:       9
```

Figure 8. Screen Shot of "Results of Data Validation"

CONCLUSION

The data validation system is necessary for researchers, database administrators and analysts as valid data is important to get the correct and reliable results from the data. The task of data validation could be enormous and time consuming to do manually, especially for a large data file. Manual validation may not be as accurate as the automated process. The data validation system can provide useful information about validation problems in a data set. The system does data validation by checking the inherent and acquired properties of the data objects. The system can check several things including invalid data type, invalid values based on the type, and out of range values. The system is standalone, user friendly, menu driven and easy to use without any specific technical knowledge. The system is going to be an invaluable tool in data analysis.

REFERENCES

[1] Bell Doug. *Software Engineering: A Programming Approach.* Addison Wesley, 2000.

[2] Booch Grady. *Object Oriented Analysis and Design with Applications*. The Benjamin/Cummings Publishing Company, Inc., 1993.

[3] Borland Software Corporation. *Borland Delphi for Windows.* Scotts Valley, CA.

[4] Budd Timothy. *An Introduction to Object-Oriented Programming.* 3rd Edition. Addison Wesley Publishing, 2001.

[5] Cantu Marco. *Mastering Delphi 7*. SYBEX Inc., 2004.

[6] Coad Peter and Yourdon Edward. *Object Oriented Analysis*. Prentice Hall, 1990.

[7] Dersham Herbert and Jipping Michael. *Programming Languages: Structures of Models*. 2nd Edition. PWS Publishing Company, Inc., 1995.

[8] Jalote Pankaj. *An Integrated Approach to Software Engineering*. 3rd Edition. Springer, 2005.

[9] Koffman Elliot. *Turbo Pascal. 5th Edition*. Addison Wesley Publishing, 1995.

[10] Kovach Warren. *Delphi 3: User Interface Design.* Prentice Hall, Europe, 1998.

[11] Merriam Sharan B. "What Can You Tell From an N of 1?: Issues of Validity and Reliability in Qualitative Research." QUIG Interdisciplinary Qualitative Studies Conference Announcement. The University of Georgia. http://www.coe.uga.edu/quig/merriam93.html.

[12] Rachele Warren. *Learn Object Pascal with Delphi*. Wordware Publishing, Inc., 2001.

[13] Thurrott Paul, Brent Gary, Bagdazian Richard, and Tendon Steve. *Delphi 3 SuperBible*. Watt Group Press, 1996.

[14] Williams Shirley and Walmsley Sue. *Discover Delphi Programming Principles Explained.* Addison Wesley Publishing, 1998.

[15] Moore Anthony. "ASP.Net Validation in Depth." Microsoft Corporation. October 2000, Updated March 2002. http://msdn2.microsoft.com/en-us/library/aa479045.aspx#aspplusvalid_rules.

[16] Microsoft Corporation. "Data Validation." http://msdn2.microsoft.com/en-us/library/aa291820(VS.71).aspx, 2008.

[17] Davis Rob. "What is Data Validity?" http://www.robdavispe.com/free2/software-qa-testing-test-tester-2214.html.

[18] Henrichsen Lynn, Smith Michael T., and Baker David S. "Validity." Research Methods in TESL and Language Acquisition. Department of Linguistics and English Language. Brigham Young University, UT, http://linguistics.byu.edu/faculty/ henrichsenl/ResearchMethods/RM_2_18.html