

Double Entry Data Capture as an Alternative Solution for Quality of E-Health Records

Vitalis Ndume

NM-AIST, Arusha, P.O.Box 447,
Tanzania,

Yaw Nkansah-Gyekye

P.o.Box 447 Arusha

Jesuk Ko

Department of Healthcare
Management, Gwangju
University

ABSTRACT

This paper discusses the techniques designed for verification of data quality in a double entry system as a part of e-health data integration. The module discussed is the subsystem of the proposed integration architecture for data sharing in a low bandwidth environment. Our algorithm encompasses methods for cleaning databases containing dirty information about household survey as well as verification of the statistical data of the household. The algorithm was applied on the double entry system for Sample Vital Registration with Verbal Autopsy and found to be generic enough to be adopted in other similar surveys which use paper work.

General term

Software Development

Keywords

Double entry, Data quality, Batch algorithm

1. INTRODUCTION

Even though there is high advancement in devices for electronic data collection there are some circumstances where use of paper work is inevitable [1-3]. These are due to the availability of enabling services and facilities such as electrical supply to electronic devices, computer literacy, training required and time, level of knowledge ranging from novice to advanced enumerators [4]. It is claimed in [5] that paperwork may be adopted due to technology adoption theory. Early adoption of electronic data collection may result in inefficiency of EDC. In addition, accidental damage of devices at the point of care and complexity of interface graphic design of the EDC software significantly affect the choice of data collection [6]. Also, unavailability of the Internet for online data collection, the speed of Internet connectivity as well as physical security of hand held devices contribute to decision of methodology of data collection [7]. Sometimes the barriers for EDC are associated with the culture. The survey conducted by [8] indicates that 60 percent of respondents thought that it was never appropriate to enter data into an EDC system in front of patients. The comparative study [9] on mobiles data entry indicates that the speed of conventional pen and paper data entry were faster than Tablet PC, d-pen and Personal Digital Assistance (PDA), even though its accuracy is low. Data entry implies a similar but rigid process with a computer intermediary which dictates certain feature rules in order for the data to be accepted by the computer.

Conducting a household survey in the developing world, both in large national survey and in small surveying research or monitoring and evaluation activity can require greater effort and considerable compromises in methodology of data collection [10]. Both EDC and paper work may be considered in parallel. In some cases, the need for collecting data with paper arise due

to coverage of the project, emergence of the epidemic in health, budget constraint, and culture change required to embrace the technology [3]. It is indicated in [6, 11, 12] that acuteness to the problem, time limit, user previous experience as well as procurement, and development of the software affect implementation outcome. While the use of EDC overweighs those of paper work the decision on to which tool to use in data collection may be influenced by the goal of automation, financial, staff, work flow constraints typically preclude simultaneous implementation of complete system [13].

With the increase of health research and even the advancement in technology, an abundance of data is collected manually and hence requires computer operators' data entry.

In a clinical system, paper work may result due to user dissatisfaction or little involvement in design [14]. Further, it is noted that reasons for health data collection manually include disaster management, conducting household survey in large national survey or season of data collection [10, 15]. Such a situation requires greater effort, considerable and compromise evaluation activities. It is argued in [7] that despite the recent effort at EDC in the industries, ongoing difficulties continue to be with collecting data in a way that does not interface with patient interviews. Although the technology is on the fore front, sometimes people prefer old fashion because it works better for them [4, 16]. It is acknowledged in [17] that it is essential to state at the outset, however, that information technology (IT) is not a panacea and that it makes some things better and others worse depending on the environment. In the field of health, clinicians often find that EDC is impractical and actually hindering the work because they are used to the ease of keeping notes on paper. There is a claim that there is less concentration when using EDC to collect data at the point of care especially for intermediate users who have less experience with digital data collection [14].

It is maintained in [9] that the intermediate step of transforming data from paperwork to electronic computer entry taking place later may result in human error. Generally, data entry personnel can be used as a validation prior to submission to the central database but it can also introduce data entry errors. Data entered in single entry normally has higher error rate and substantially may harm data analysis and interpretation [13]. Furthermore, the algorithm for double entry system requires being precise enough to detect error in spelling or omission. If errors can be detected before analysis then a decision drawn from the data can be trusted for building strong health policy [17]. Errors in data that are collected manually are frequent because they may result from the whole cycle of processing the document from the point of care to the data entry. While most errors have little potential for harm, some do result in injury and the cumulative consequences is huge. Therefore, it is necessary to have an effective method for early detection of error in the

source data collected manually before analysis. Even though it increases time of data entry operator, double entry is most preferred in data cleansing [18].

This paper describes the batch algorithm developed for double entry data captured using a baseline census of vitals registration with verbal autopsy as case study. The module discussed is the subsystem of the proposed integration architecture for e-health in a low bandwidth environment. The paper also presents a method for discovering dirty information patterns from data. Our algorithm encompasses methods for cleaning databases containing dirty information about household survey as well as verification of the statistical information in the hierarchy of base information. The proposed algorithm was implemented and found to be accurate enough to improve quality of data. Moreover, it was generic to be adapted to other systems for quality control in double entry system. The nomenclature for naming data integrity was deduced from the known national code, that is, the region code two digits, district code two digits, ward code three digits, enumeration areas three digits, household code four digits and member code two digits.

1.1 Data Quality Problem

Data cleaning is necessary in order to increase quality of data. Data cleaning is especially required when entering heterogenous data sources and should be addressed together with schema-related data transformation [19]. Furthermore, data cleaning deals with detecting, reporting and removing errors and inconsistency from data in order to improve the quality of data. There are many methods in data cleaning including visual assessment, auditory, use of computer programs, etc.

Data quality problems are inevitable in single data entry due to misspelling during data entry, missing information or other invalid data. The classification of data quality problem is described in Figure 1. There are many sources of data errors when recorded manually, e.g., no value recorded on paper sheet, handwriting recognition, data recorded out of box, values recorded are out of the range, the system does not recognize the value, etc [13]. When multiple data sources need to be integrated into a data warehouse or in a web based information system, the need for data cleaning increases significantly [19]. The situation is more required when verification is not done at

the point of care. This is because the sources often contain redundant data in different representation in order to provide access to accurate and consistent data, consolidation of different data representation and elimination of duplicate information becomes necessary. Due to the wide range of possible data inconsistencies and the sheer data volume, data cleaning is considered to be one of the biggest problems in data inconsistencies. A large number of tools of varying functionality is available to support task of data cleaning but often a significant portion of the cleaning and quality control in a single entry is done manually or by low-level programming that is difficult to write. When using multi methodology of data collection in a joint project or in a large project, data is not directly integrated into a data warehouse before cleansing but needs to be extracted from multiple sources, transformed and combined to enable data analysis be performed easily and query them from single source.

Even though human errors in health care system have only recently received great attention, human factor engineering has been concerned with errors for several decades. It is claimed in [17] that while it is easy and common to blame operators for accidents, investigation often indicates that an operator “erred” occurs because the system was poorly designed. Data entry errors can be disastrous. Data need to be of quality to facilitate decision making. In many industries, data entry error is minimized by employing skilled data operators or by using higher technology which is not affordable to small businesses or non-profit organizations. However, this results in increased cost of the project [20].

It is emphasized in [9] that the adoption of web-based data entry system which introduces real-time verification and submission has not resulted in a significant gain in terms of data occurrence or efficiency. The problem is associated with the unavailability of on-line data accuracy checks as well as reliability of the network. Many specialized cleaning tool typically deals with a particular domain, mostly name and address or concentrates in eliminating duplication. An algorithm that goes beyond just removes duplication is therefore important in data quality problems. As indicated in Figure 1, the complexity of the data quality arises from two sources: single source data entry or multi-source data entry.

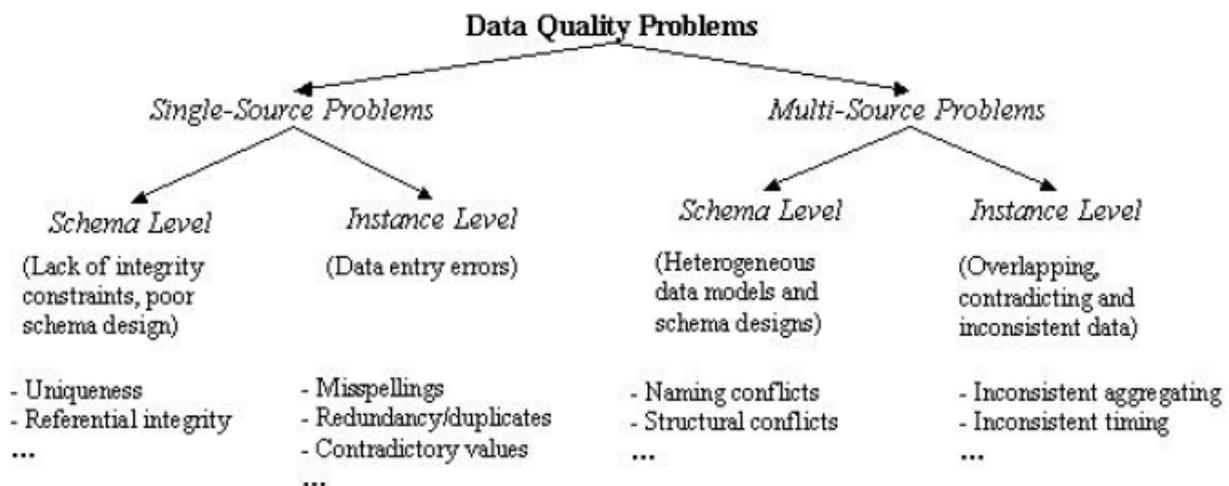


Figure 1: Classification of data quality problem in data sources (Rahm and Do, 2000)

2. METHODOLOGY

A survey method was deployed to collect data by paper from four regions where one district was selected from each region. The National Bureau of Statistics (NBS) in Tanzania was responsible for sampling of the enumeration area. The sampling technique method worked out as follows:

The country was divided into 8 zones. The NBS uses probability proportion [21] to obtain 3 districts per zone. Each district was selected from a region. As a result 23 districts were sampled for SAVVY project estimated to 21, 000 households per zone making a total of 168,000 households nationwide. The NBS applied probability proportional to obtain enumeration area (EA) per district. As a result 10 EA per district, about 700 households per district and around 35,000 populations were estimated. As per NBS specifications EA takes a maximum of 60 households but due to recent immense increase of population in Tanzania, this number has increased up to more than 100. All households were enumerated within selected EAs. That is about 161,000 households and over 805,000 population coverage nationwide. Other important aspects considered include: All EAs were chosen from Tanzania Main-land only; national estimation was based on the 2002 census, equal representation of strater from rural and urban representation.

Data integrity was assured by developing a unique identification from the national coding system. The nomenclature for naming follows the following patterns: region code two digits, district code two digits, ward code three digits, and enumeration area three digits as shown in Table 1 below. The last two naming that is household code was 4 digits and member ID was 2 digits. The first four hierarchical orders of naming and code were generated from National Bureau of Statistics (NBS) and the last two naming was proposed by the research team. The software was developed to capture these last digits at the first entry. A double entry algorithm was developed where key identification for data mapping was deduced from the above nomenclature.

Table 1: Unique Identification of the record

Region Code	District Code	Ward Coed	Enumera tion Area	Househo Id Code	Member Id
06	01	021	011	0001	01

During the data entry, the system generates the household code in a First Come First Serve (FCFS) algorithm [22]. The household code was therefore concatenated with the NBS nomenclature code. Similarly, the household members' identification was generated. The member code was estimated not to be more than 100 per household and therefore two digits were enough to handle the code. At the same time, household per EA was expected to be in a digit of '000' and therefore four digits was enough to record all households in one EA.

2.1 Physical Identification of Household

For each booklet recoded for enumeration, the supervisor stamps each document in the booklet using 8 digit stamps. Visual document number was therefore generated by combination of household number and serial stamping number. These approaches uniquely identified the records visually. In a database system this physical identification was concatenated with record identity to produce composite keys.

The system was developed using Visual Studio 2009 where Visual Basic Dot Net was a programming language and SQL server 2005 database was configured for storage. A local network was built and Server client architecture was set up at Dar es Salaam Institute of Technology where a total of 35 PCs were connected in the computer laboratory. Data entry operators were hired by the SAVVY project and trained for one day before commencing the activity.

3. RESULT AND DISCUSSION

Data entry software was developed and implemented. The software was used to convert data collected manually to electronic format. The package was embedded with double entry algorithm. This software overcomes some validation and verification challenges which are common to data analysis team and hence ensures the quality of data before archiving to a staging database. The baseline census data for sample vital with registration with verbal autopsy was used for implementation. A total of 33,505 sampling of households from four regions was surveyed and recorded in a hard copy booklet. Out of that 148,918 family members were registered. In addition, 4382 deaths and births events were recorded. The average household members were found to be 4.4465 approximated to 5 members per household.

3.1 Algorithm to Compare Data

In the batch algorithm, two data tables that hold data to be compared need to be available at once and that the value of the existing data table which is assumed to be correct is replaced (if any) and stored at the end of the operation. This implies that for database *D*, we need corresponding *D'* of the same to exist as a file or temporary during run time.

```

1 Input : Data table 'A' containing existing values
2   Data table 'B' containing Current values
3 Output: Correct values as per source documents
4 Start
5 Load data table A containing existing record
6 For each tuples in binding of data table A
7   Load data table B containing current records
8   Filter the primary key corresponding to current tuple
9   Compare two values
10  If mismatch is detected Display these values
11    Confirm the correctness from source booklet
12    If value of the column in data table B is correct
13      Replace value of data table 'A' with value of data datable 'B'
14    Else
15      Retain the value of data table 'A'
16    End if
17  End if
18 Save and store data table 'A'
19 Loop
20 End
  
```

Figure 2: Pseudo code for comparing two data of the data table

The approach was to load existing data on the data table and current information entered was also loaded on another data table of the same database. The primary key was used to ensure integrity.

The pseudocode that describes the algorithm is presented in Figure 2.

The class function that implements the above algorithm is presented in Figure 3 below. The input to the algorithm is the records of the existing values and the current values entered by data operator. The trigger of the comparison is done by entering physical identification of the document.

In the double entry and specifically in this case three relations were subject to verification. Other entities were pre

populated with data as a drop down list or as lookup table during the data entry. In addition, there were some variables/items within a relation which were conditional and therefore were not excluded from verification.

Even though asymptotic analysis is the standard benchmark condition for algorithm analysis, other means of analysis such as machine speed, programming language and number of codes as well as compiler can be used. The batch algorithm was implemented with 66 lines of codes. Space efficiency was reasonably considered. The primary consideration when estimating an algorithm performance is time efficiency; this depends on the number of basic operation required by the algorithm to be processed and input size [23]. In this case the number of size depends on the arrays string that the user keys in and also the number of column items that the system needs to iterate through. That means for each record **R** it loops through column **C** and for each column it picks the value item V_1 and iterate through it and compare it with the current value V_2 . Each value in V_1 is compared with each value in V_2 and a mismatch is reported if any. An operator therefore uses visual assessment to accept or reject the changes. The information in the source

booklet was used as visual control when mismatch or error is reported.

The outer loop considers four relations of which action was triggered by user events (the outer loop is not shown). The inner loop (**line 6**) considers number of items/ columns in the relation. To avoid the third inner loop, the inbuilt functions were used that converted the column values V_1 and V_2 into upper case and used linear operation, that is, conditional operation. This approach reduces the time of looping to a constant time C_1 which has no match effect and therefore taken as zero. These approaches increase efficiency of the algorithm. The assumption made during run time is that the existing record is taken as legitimate input and the current record is taken as a verifier input. The two inputs are therefore compared. The operation in **line 7** excludes columns that need not be compared while **lines 8** and **9** check for null values in existing record and current record. **Line 19** checks if required comparison is for string values and the string comparison is done at **line 20**. **Lines 22 - 28** report mismatch of the string operation. **Line 40 to 55** compare non string objects and reports the error message to the user for decision.

```
1 Public Sub CompareData(ByVal CurrentBindingSource As Windows.Forms.BindingSource)
2   If _MyExistingBindingSource.Count > 0 Then
3     If CurrentBindingSource.Count > 0 Then
4       Dim CurrentDr As DataRowView = CurrentBindingSource.Current
5       Dim ExistingDR As DataRowView = _MyExistingBindingSource.Current
6       For Each col As DataColumn In _MyExistingDataTable.Columns
7         If ExcludingColumns.Contains(col.ColumnName) Then Continue For
8         If ExistingDR(col.ColumnName) Is DBNull.Value Then
9           If CurrentDr(col.ColumnName) Is DBNull.Value Then
10            Continue For
11          Else
12            GoTo Mymsg
13          End If
14        Else
15          If CurrentDr(col.ColumnName) Is DBNull.Value Then
16            GoTo Mymsg
17          End If
18        End If
19        If col.DataType Is GetType(String) Then
20          If ExistingDR(col.ColumnName).ToString.ToUpper <> CurrentDr(col.ColumnName).ToString.ToUpper Then
21            Dim V_string As String = col.ColumnName
22            If _ListOfColumnNames.Contains(col.ColumnName) Then V_string = _ListOfDisplayNames(_ListOfColumnNames.IndexOf(col.ColumnName))
23            End If
24            If MsgBox("The information:" & V_string & " Does not match." & vbCrLf & _
25              "The existing is:" & IIf(ExistingDR(col.ColumnName) Is DBNull.Value, Nothing, ExistingDR(col.ColumnName)) & _
26              "Do you want to replace:" & IIf(ExistingDR(col.ColumnName) Is DBNull.Value, & _
27              "", ExistingDR(col.ColumnName)) & " to be " & IIf(CurrentDr(col.ColumnName) Is DBNull.Value, & _
28              "", CurrentDr(col.ColumnName)) & "?", MsgBoxStyle.YesNo + MsgBoxStyle.Information) & _ = MsgBoxResult.Yes Then
29              Else
30                CurrentDr.BeginEdit()
31                CurrentDr(col.ColumnName) = ExistingDR(col.ColumnName)
32                CurrentDr.EndEdit()
33            End If
```

```

34 End If
35 Else 'Non string entry
36 'check for data value of its property is not string
37 If ExistingDR(col.ColumnName) <> CurrentDr(col.ColumnName) Then
38 Mymsg:
39 Dim V_string As String = col.ColumnName
40 If _ListOfColumnNames.Contains(col.ColumnName) Then
41 V_string = _ListOfDisplayNames(_ListOfColumnNames.IndexOf(col.ColumnName))
42 End If
43 Dim myExistingvalue As String = IIf(ExistingDR(col.ColumnName) Is DBNull.Value, & _
44 "", ExistingDR(col.ColumnName))
45 Dim myExistingValue1 As String = IIf(ExistingDR(col.ColumnName) Is DBNull.Value, & _
46 Nothing, ExistingDR(col.ColumnName))
47 Dim myCurrentValue As String = IIf(CurrentDr(col.ColumnName) Is DBNull.Value, & _
48 "", CurrentDr(col.ColumnName))
49 If MsgBox("The informaion:" & V_string & " Does not Match." & vbNewLine & _
50 " The Existing Information:" & IIf(ExistingDR(col.ColumnName) Is DBNull.Value, & _
51 Nothing, ExistingDR(col.ColumnName)) & _
52 " Do you want to Replace:" & IIf(ExistingDR(col.ColumnName) Is DBNull.Value, & _
53 "", ExistingDR(col.ColumnName)) & "to be" & IIf(CurrentDr(col.ColumnName) Is DBNull.Value, & _
54 "", CurrentDr(col.ColumnName)) & "?", MsgBoxStyle.YesNo + MsgBoxStyle.Information) & _
55 = MsgBoxResult.Yes Then
56 Else
57 CurrentDr.BeginEdit()
58 CurrentDr(col.ColumnName) = ExistingDR(col.ColumnName)
59 CurrentDr.EndEdit()
60 End If
61 End If
62 End If
63 Next
64 End If
65 End If
66 End Sub

```

Figure 3: The batch algorithm for data validation

3.2 Double Entry System as Sub System for Data Integration

There were three sources of data in the project as illustrated in Figure 3. The first are those collected electronically. These were pushed to the staging database using web service technology. The second source of the data was those which were collected using paper. These were treated in two ways: data were first entered by data entry operators in a single entry. Then the supervisors were responsible to verify the data using double entry algorithm in as shown in Figure 3 above. Two types of verification were possible:- First, statistical verification which reports the number of family members in a household number of events that is birth and deaths. Second, correctness verification which is concerned with quality, accuracy and consistency of the data values.

The architecture demonstrates the data entry as sub system in the process of digitizing data surveyed by paperwork before archiving into the staging data. The collected data must be validated and verified before analysis. The process therefore requires software which can improve the quality of data. The double entry system is mostly used to ensure that data entered in the first time is error free. Once validated and verified, they are pushed to the staging database using merging module. The stored procedure and API was developed to merge data to the staging database.

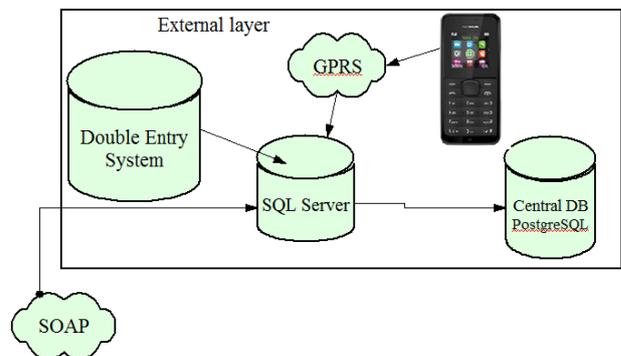


Figure 4: Proposed architecture of the subsystem for double entry

3.3 Concurrency Control in Record Identification in a Slow Network

The easiest way to develop identification for a record is to generate record identification on the client side and then the whole complete records are transferred from client application to server for storage. Even though, this approach is faster in a slow network but it generates concurrency. Double entry set up requires generating the record identification in the server side by doing so a concurrency control can be controlled because there will be no duplication of records. In order to solve this problem, a store procedure was developed in order to generate the Household ID from the server side.

3.4 Interface Design

Data to be compared require a three step interface design. The interface in Figure 5 shows the basic information at the first stage of double entry system. It is recommended in [6, 24] that the most accessible interface for database must be intelligent and able to understand the natural language expression as well as a vague query. This interface allows one to impose a semantic query with *WHERE* clause. When the user keys in the document number and then presses the *OK* button, the query is formulated to retrieve basic information that enables the user to understand the records he/she needs to compare. Once basic

information is compared and saved then the user proceeds to compare details of the members by clicking the **Next** button.

The screenshot shows a software interface titled "SECTION A: IDENTIFICATION". It contains several input fields and buttons. The fields are: Region (DAR ES SALAAM), District (Kinondoni), Ward (Mabibo), E_Area (121 Jitegemee), Document Number (07), Round (1), Household Name, HouseHold ID (0022), Hamlet Leader, Location, House Type, Number of Visit, Date of Visits, Interview Date, Total Number of Household, and Latitude. There are buttons for "OK", "Back", "New", "Save", "Delete", and "Next". The interface is divided into four sections: SECTION A: IDENTIFICATION, SECTION B: LOCATION DESCRIPTION, SECTION C: PREVIOUS VISITS, and SECTION D: INTERVIEW RESULTS.

Figure 5: Interface for Basic information

4. CONCLUSION

Data collection is an important aspect of any type of research study; however, inaccuracy in data collection can impact the result of the study and ultimately lead to an invalid decision. The software package design is of utmost consideration to data quality improvement. Access to good quality data, latest national and regional data on disease burden, needs assessment or intervention coverage is essential to help informed national policy decision; as well as guide relief effort, monitor control and evaluate their impact, progress and cost effectiveness.

This package software system is a first step in designing a dynamic link library that can be used for rigorous evaluation of data quality and valid class that can be adaptable for other double entry systems. The idea behind this software is to find a solution for quality of data collected using paper work. We have designed an algorithm that cleanses data by eliminating duplication, spelling mistakes, and therefore increases accuracy of data. The generality of this algorithm and its implementation makes it an attractive method for new application in ensuring data quality in double entry system.

5. REFERENCES

- [1] D. Babre, "Electronic data capture—narrowing the gap between clinical and data management," *Perspectives in clinical research*, vol. 2, p. 1, 2011.
- [2] A. Maedche and S. Staab, "Ontology learning for the semantic web," *Intelligent Systems, IEEE*, vol. 16, pp. 72-79, 2001.
- [3] B. Cherry, *et al.*, "Factors Affecting Electronic Health Record Adoption in Long-Term Care Facilities," *Journal for healthcare quality*, vol. 30, pp. 37-47, 2008.
- [4] A. L. Terry, *et al.*, "Adoption of electronic medical records in family practice: the providers' perspective," *Family medicine*, vol. 41, p. 508, 2009.
- [5] C. Barbara, *et al.*, "Experiences with electronic health records: Early adopters in long-term care facilities," *Health care management review*, vol. 36, pp. 265-274, 2011.
- [6] A. Motro, "VAGUE: A user interface to relational databases that permits vague queries," *ACM Transactions on Information Systems (TOIS)*, vol. 6, pp. 187-214, 1988.
- [7] MI-CO. (2008, Beyond EDC: Resolving Data-entry Challenges faced by Electronic Data Capture Systems in Clinical Trials. *White paper*. Available: www.beyondedc.com
- [8] S. Redfarm. (2009, More paper More Burden: White paper. Available: http://www.clinpage.com/article/more_paper_more_burden/C5 accessed on March 2014
- [9] E. Cole, *et al.*, "A comparative study of mobile electronic data entry systems for clinical trials data collection," *International Journal of Medical Informatics*, vol. 75, pp. 722-729, 2006.
- [10] J. L. Vanden, *et al.*, "Use of handheld computers with global positioning systems for probability sampling and data entry in household surveys," *American Journal of Tropical Medicine and Hygiene*, vol. 77, pp. 393-399, 2007.
- [11] D. A. Ludwick and J. Doucette, "Adopting electronic medical records in primary care: lessons learned from health information systems implementation experience in seven countries," *International Journal of Medical Informatics*, vol.78, pp. 22-31, 2009.
- [12] B. Framian, Pavel Salz, " Definition of Data Collection Needs Aquaculture " 2009.
- [13] P. C. Dykes, *et al.*, "The feasibility of digital pen and paper technology for vital sign data capture in acute care settings," in *AMIA Annual Symposium Proceedings*, 2006, p. 229.
- [14] J. E. Stevenson, *et al.*, "Nurses' experience of using electronic patient records in everyday practice in acute/inpatient ward settings: a literature review," *Health Informatics Journal*, vol. 16, pp. 63-72, 2010.
- [15] R. Krishnamurthy, *et al.*, "Application of pre-programmed PDA devices equipped with GPS to conduct paperless household surveys in rural Mozambique," in *AMIA annual symposium proceedings*, 2006, p. 991.
- [16] A. L. Terry, *et al.*, "Implementing electronic health records Key factors in primary care," *Canadian Family Physician*, vol. 54, pp. 730-736, 2008.
- [17] D. Bates, *et al.*, "Reducing the frequency of errors in medicine using information technology," *Journal of the American Medical Informatics Association*, vol. 8, pp. 299-308, 2001.
- [18] R. A. Reynolds-Haertle and R. McBride, "Single vs. double data entry in CAST," *Controlled clinical trials*, vol. 13, pp. 487-494, 1992.
- [19] E. Rahm and H. H. Do, "Data cleaning: Problems and current approaches," *IEEE Data Eng. Bull.*, vol. 23, pp. 3-13, 2000.
- [20] K. A. Barchard and L. A. Pace, "Meeting the challenge of high quality data entry: A free double-

- entry system," *International Journal of Services and Standards*, vol. 4, pp. 359-376, 2008.
- [21] N. A. Weiss and C. A. Weiss, *Introductory statistics*: Pearson Education, 2012.
- [22] M. de la Mata, *et al.*, "First come, first served revisited: factors affecting the same alternative splicing event have different effects on the relative rates of intron removal," *Rna*, vol. 16, pp. 904-912, 2010.
- [23] c. A. Shafter, *Data structures and algorithm analysis in java 3rd Edition*, 3 ed.: Dover Publications, 2012.
- [24] N. Nihalani, "An Intelligent Interface for relational databases," *human-computer interaction*, vol. 6, p. 7, 2010.