SOFTWARE ENGINEERING: A FUNDAMENTAL APPROACH TO AUTOMATE AND INTERFACE MEDICAL ELECTRONIC SYSTEMS WITH COMPUTERS IN HEALTHCARE SYSTEMS

Hashim M. Twaakyondo and Joseph E. Mbowe
Department of Computer Science, University of Dar es Salaam
P.O. Box 35062 Dar es Salaam
e-mail: htwaaky@udsm.ac.tz, hmtwaaky@yahoo.com

ABSTRACT

*The cost for application software and propriatry standards (i.e. data structure and formats) required for storing clinical datasets from electronic device cause the use of time-consuming paper-based documentation and/or the transfer of electronic lab records manually to the PC database systems in Mother-Offspring Malaria Study (MOMS) Laboratory Unit located in Morogoro Regional Hospital, Tanzania. We have explored the ability to extract laboratory results from Haematology analyzer machine for easy management, access and storage. The methodology and tools used during the study includes; site survey and hands-on practice for requirement engineering, visual basic 6.0 for interface design and Microsoft Access for relational database design. The study has developed a model "**CO**mpose-**TRA**ansmit-**CO**nvent-**CO**mpose" convention (acronym COTRACOCO) to implements an open serial port interface that required only a driver class to execute the labs outputs for storage in database with at run-time with less programming knowledge. The study demonstrated that the COTRACOCO convention presented is general and may promote a new paradigm for establishing the robust serial port communication systems for any digital device*

***Key words :** COTRACOCO , Serial Port, LIMS , MOMS, PC Database systems and ABX Micros 60 OS/OT.*

INTRODUCTION

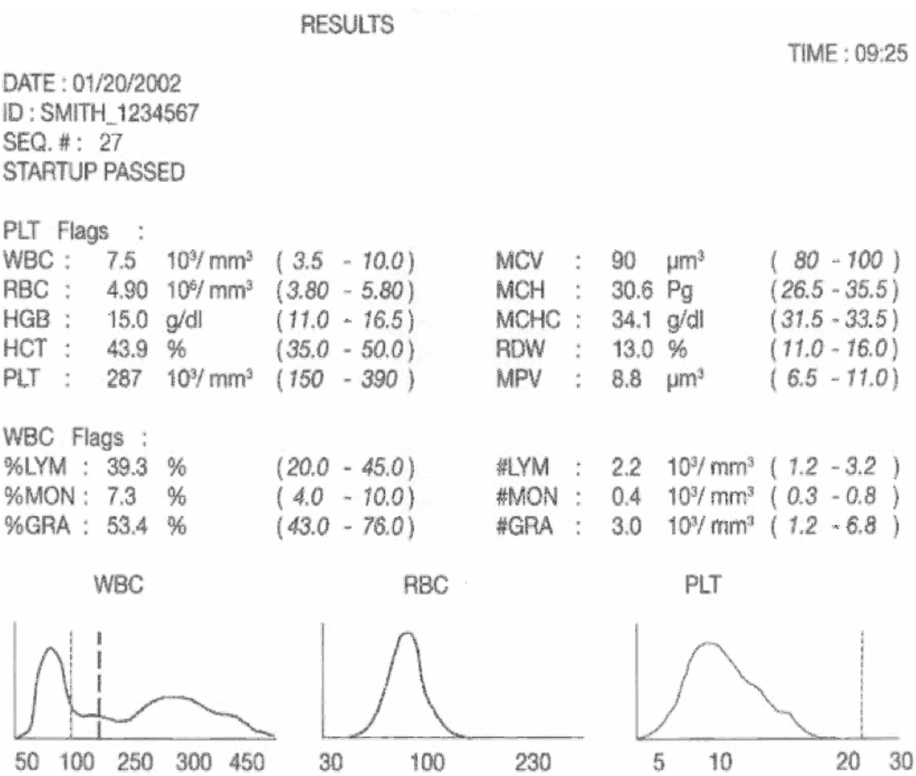
The advent of micro computing early 1980's has promoted interest in the use of computers in medical practice (Cooper et al, 1989; Stock et al, 1985; **Markivee, 1985**; Ellis et al, 1987). Also, the revolutionary of electronics' industries like ABX Diagnostics LTD has promoted development of new ICT tools and powerful automated electronic devices (ABX, 2003) in medical systems. Reinvention rules (Coiera, 2004) for laboratory automation and steps required for the validation of Laboratory Information Management Systems i.e. LIMS (**Turner et al, 2001**) provide a clear path for development of robust healthcare systems. The important benefits of LIMS (Nelson, 1969) in health care systems; critically provides a very challenging scene for developing a modern tools for capturing and storing complete blood count (i.e. from Micros 60 OS/OT) for future analysis and retrieval and provide behavioural understanding of complex systems for better healthcare system design. Although , there exist PC applications to read from serial ports, most of serial port application have proprietary standards and does not organise and store the patient results in PC database (ABX, 2003), except some systems can generate the results in restricted file-formats in accordance with the automated elecronic devices; the good example is Hemalink software used to read results from Micros 60 OS/OT which has only ability to store the clinical datasets into XMLs data formats in MOMS laboratory unit. It's time-consuming for simple querying and/or writting scripts or to convert XML's dataset contents into necesary patient records required by clinical officers for patient medication purposes.

The Personal Computer (PC) and Database Management Systems (DBMS) avaliable nowdays provide tools for organising flat files (i.e. text based or indexed files) into relation or records of the same data formats. The recently use of relational database systems i.e. RDBMS allow development of new applications to store data into tables (i.e. mySQL , SQL server, Oracle). In order to store the clinical datasets from Micros 60 OS/OT machine at run-time, system analysts and programmers should be able to write serial port programs to read and load clinical datasets into PC local database for fast data mining and access. All digital devices communicate using the stream of zeros and ones to exchange their information, this idea provide a fundermental concept to collect the stream of bits transmitted into Character-Array basement.

By using bits-collection approach , then; the time, effort and proprietary standards applied when developing serial communication systems can be reduced by implementing an open model or method that would read the sequence of bits transmitted by two communicating device and store them onto an array system for future extraction of specific data blocks or records as may be determined by system users. The method should not require any prior information for conection protocol; except the collected bits may be compared from the printouts of such device to provide means to extract the required data block for the purpose of determining certain part of patient information. The general model to convent the bitstream from electronic systems should be copyleft inorder to facilitate the code-reusability for new modification; public downloads and configuration to support data loading from electronic devices to PC database systems.

The new method should be able to address and solve the questions like: (1) Can the specifications be understood by normal user? (2) How difficult is it to reconfigure for new applications? (3) Can data be transferred easily to new applications? (4) How friendly the user interface ? (5) How easy and inexpensive is it to modify? (6) Which application packages are supported? (7) Is the package copyright or copyleft? .

Generally, the copyleft systems nowdays are important in IT industries (i.e. especially developing countries with less or poor budget in IT) just to facilitate open mind and code re-usability for not re-inventing the wheel. The solutions to address the above questions created a great challenge to develop a new technique or method that would compose and transmit the stream of bits into collection of characters or string. The existing Hemalink application, has ability to produce the patient information from Micros 60 OS/OT by printing the values into paper-based form as shown in Figure 1.0 at run-time. The use of paper-based information cause poor information searching and system efficiency. The softcopy of the printed datasets contents are stored in XMLs file formats , whereas , not all users can process and access XMLs files more confortably. In order to facilitate the automation of smart devices used in MOMS Labs to support relational database for fast data querying and storage, the study has developed a model "**CO**mpose-**TRA**ansmit-**CO**nvent-**CO**mpose" convention (acronym COTRACOCO) to implement serial port interface to compose and transmit the stream of bits (i.e. byte that leads to character) for storage in relational database with less programming knowledge.


RESULTS

TIME : 09:25

DATE : 01/20/2002
ID : SMITH_1234567
SEQ. # : 27
STARTUP PASSED

PLT Flags :

WBC :	7.5	10^3/ mm^3	(3.5 - 10.0)	MCV :	90	µm^3	(80 - 100)
RBC :	4.90	10^6/ mm^3	(3.80 - 5.80)	MCH :	30.6	Pg	(26.5 - 35.5)
HGB :	15.0	g/dl	(11.0 - 16.5)	MCHC :	34.1	g/dl	(31.5 - 33.5)
HCT :	43.9	%	(35.0 - 50.0)	RDW :	13.0	%	(11.0 - 16.0)
PLT :	287	10^3/ mm^3	(150 - 390)	MPV :	8.8	µm^3	(6.5 - 11.0)

WBC Flags :

%LYM :	39.3	%	(20.0 - 45.0)	#LYM :	2.2	10^3/ mm^3	(1.2 - 3.2)
%MON :	7.3	%	(4.0 - 10.0)	#MON :	0.4	10^3/ mm^3	(0.3 - 0.8)
%GRA :	53.4	%	(43.0 - 76.0)	#GRA :	3.0	10^3/ mm^3	(1.2 - 6.8)

Figure 1.0 Complete blood count values from Micros 60 OS/OT device

Implementation of COTRACOCO

The COTRACOCO convetion developed, identifies all bits that composed the single character of the entire patient data string or patient complete blood count during the transmission. The method has defined four phases or stages (Figure 1.1) to provide an insight view for any programmer to implement the tecnique so that it can capture and store the characters received from sender or bits generator into an object array system for visual display unit (VDU). The crossection as show in figure 1.1 describe two parties (1) the sender part i.e. at the left of the **T***ransmiter* and (2) the receiver part i.e. at the right of the **C***onverter*. The arrow show how bits flow from sender to receiver. For example, the bits are transmited from sender memory buffer by composing the stream of zeros and ones and reach receiver memory buffer through sender serial port before converted from memory buffer to character basement for user display at VDU through receiver serial port.

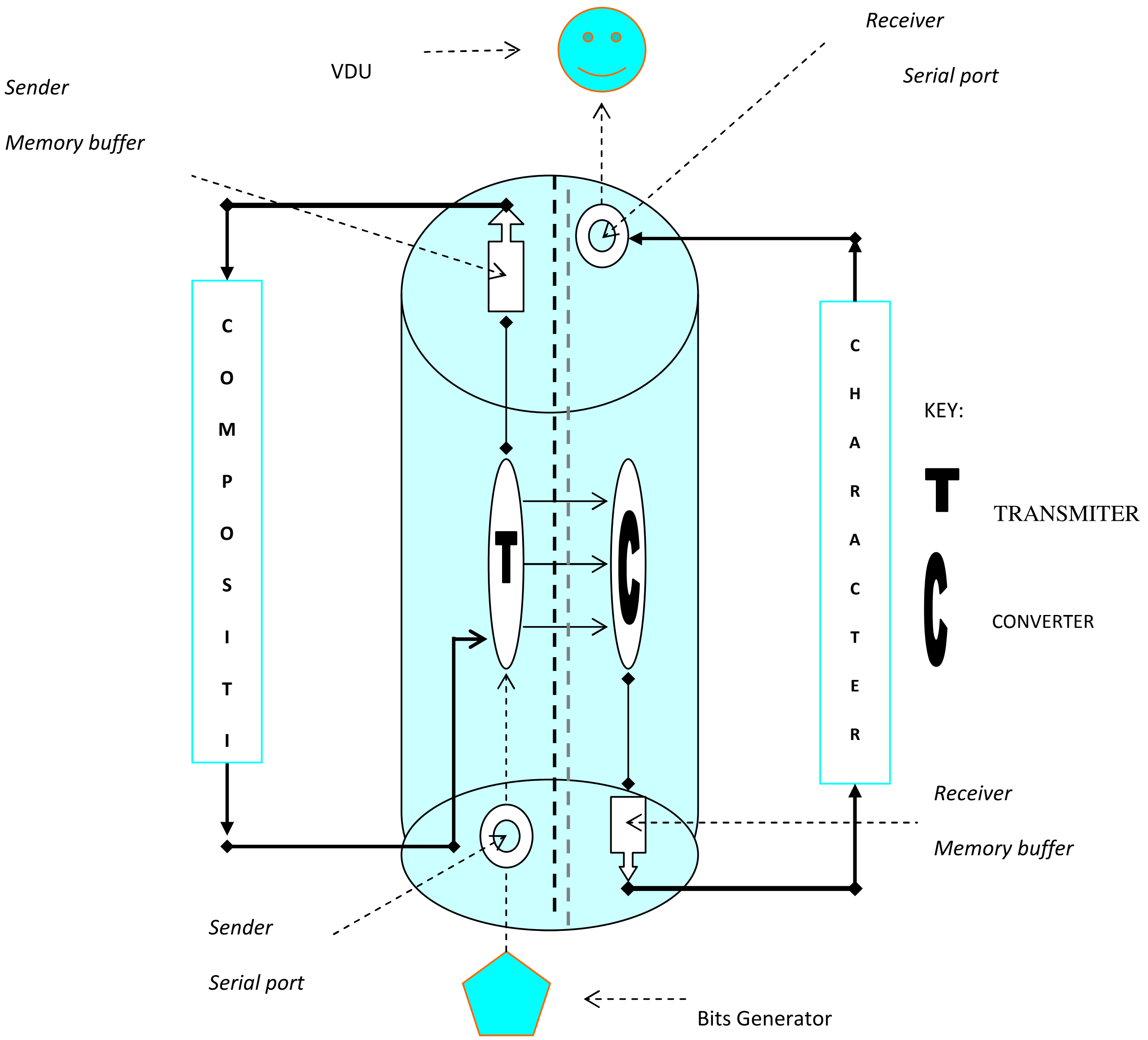


Figure 1.1 COTRACOCO insight user view

In order to facilitate the fusion of computing and advances in IT for medical education (**Masys, 1998**) and usage in Information Communication Technologies (ICT), our study has discussed the methodlogies to be used for a programmer to implement the COTRACOCO technique discussed in figure 1.1

Byte-Array Composition methodology stage: all characters (i.e. the complete blood count data block) are packed into byte data type object. In any serial link communication, the serial ports transmit or receive data packed onto object of byte data type. The Listing 1.1 describe the pseodecode for packing the bits of each individual characters into byte object before transmission. This technique would provide easier transformation of byte object into variant data type object that would be managed easily at the end of the serial link communication. After packing the bits into byte object, the object is assigned to variant object ready to be written into serial port memory registers for transmission.

```
Store the bits in ByteArray
/*Declare Byte Array datatype object
ByteToSend(0)= bit0
ByteToSend(1)= bit1
…          =    …
ByteToSend(n)= bitn
/* END
```

Listing 1.1: Algorithm for bits array composition during serial communication

Byte-Array Transmission methodology stage: the RS-232 interface uses an "asynchronous" protocol during serial port communication. In this protocol, no clock signal is transmitted along the data. The receiver has to have a way to "time" itself to the incoming data bits so that it can keep the track of the sequence of the bits transmitted along the communication channel. In the case of RS-232; the following principles (Axelson, 2000) should be observed while the link is established; both side of the cable must agree in advance on the communication parameters (speed, format, parity and stop bits etc). That's done by configuring the serial port settings. The transmitter sends a "1" when and as long as the line is idle, this would take care of over-flow and data collision. The transmitter sends a "start" (a "0") before each byte transmitted, so that the receiver can figure out that data is coming. Also, since byte is collection of 8-bits that form a single character, it allows the receiver to mark the start and end of each character. After the "start", data comes in the agreed speed and format, so the receiver can interpret and convert into required data format for human redability. The transmitter sends a "stop" (a "1") after each data byte is transimitted sucessfully. At this stage , the byte object is composed of 8-bits blocks and each block represent the single character or letter transmitted at a time, the Listing 1.2 provide way of transforming and assigning the byte into variant object for transmission.

```
Write the Byte Array in variant object
/*Declare bitsBuffer as Variant
Store the bits in ByteArray
bitsBuffer = ByteToSend()
Write the variant to the port
portMemory = bitsBuffer
/* END
```

Listing 1.2: Algorithm for writing in serial ports

At the other hand, the recieiver's serial port time itself to the bits transmitted from the sender. The algorithm (Listing 1.3) described the possible steps to read the incoming data from serial port.

```
Read the Bits Received as Byte Datatype
/*Declare bitsBuffer as Variant
Read the bits from serial port
bitsBuffer = portMemory
Store the variant into Byte Array
ByteToReceive () = bitsBuffer
/*END
```

Listing 1.3: Algorithm for reading in serial ports

Bit-character Convention methodology stage: implements the methods (Listing 1.4) that convert each incoming block of eights bits into character (i.e. 8 bits compose 1 byte and 1 byte compose 1

Character) so that it can keep the track of all bits transmitted along the communication channel to represent the meaningful information for human readability compared to zeros and ones which represent numbers in a human scene. These bits are converted into characters at the run-time immediately as one byte is received, thereafter, the byte or character is stored in an indexed object for the purpose of future extraction using index searching techniques or any other method that can be implimented. The adoption of in-builtin function in VB.6 like Chr$() (**Balena, 1999**) would support the convention of bits into characters at the run-time during transmission period .i.e. while the serial port transmit the stream bits. The character, would explicitly be stored into mechanism that can be retrieved sequantially. The method use the First-In-First-Served (FIFS) technique to ensure no information is lost in the way while storing the characters.

```
Declare the getChar to handle the convention of bits
/* Declare the Array objectVariant
For i = 0 To Ubound (ByteToReceive) - 1
getChar = getChar & (Chr$(BufferArray(i)))
objectVariant [i] = getChar
Next i
/* END
```

Listing 1.4: Algorithm for converting bits into Character

Characters-Array Composition methodolgy stage: create the databank object to collect the entire collection of bits transmitted and received from serial port link. The stage implements methods with object or variable array data type in order to store all characters collected during real-time transmission. After completion of the transmission, the method also implements a two sub methods; Firstly sub-method that triggers the end of serial link transmission. Secondly, sub-method that extracts and store all specific complete blood count parameters into relational database as shown Listing 1.5. In this stages, all security measures and data encryption would be implemented at this stage during datasets loading onto PC database systems. The security include windows database authentications, the use of user priveledges as defined by system administrators and the use of encryption and decryption technology.

```
FOR ALL indexes Loop
For index = startRange To endRange
para_value = para_value &_
readObejectVariantArray(index)
Next index / *until end range*/
get_para_Value = para_value
form_object_para_variable = get_para_value
Loop until
ALLSTORE_ALL_PARAMETERS_TO_DATABASE
Check_Link_Method (t)

/* END
```

Listing 1.5: Algorithm for extraction of specific parameters From array variant object of characters

Serial port application interface and database schema

The serial port Graphical User Interface (GUI) between the user and the computer systems were developed to use a mouse to sophisticate the user's interaction with the computer. The Figure 2.0 allow user to run and stop application from listening the serial ports of communicating devices. The sequences of characters are displayed on the text-box provided in Figure 2.0 (i.e. WYIWG) before extracted into specific data block of complete blood count results as shown on Figure 2.1 (i.e. QC Results from Micros 60).

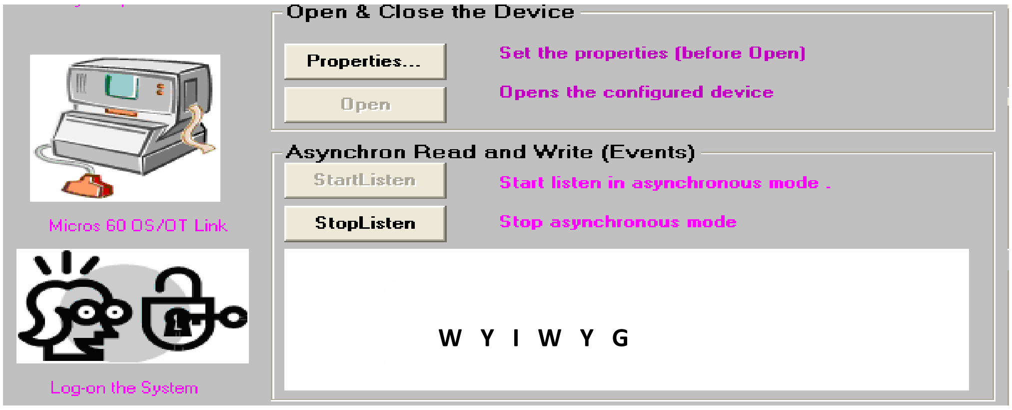


Figure 2.0 User interface to initialize the serial port settings

The patient laboratory results are stored into relational database schema by including the information such as nurses, technicians and doctors attended such patient during recruitment process. The user has option to accept order, to request new order and /or update order etc as shown on Figure 2.1. After accepting the order, the data captured in the form shown in Figure 2.1 are stored on the schema available in Figure 2.2. The table has extended to comprise the technicians, nurses and doctors attended the patient.

.

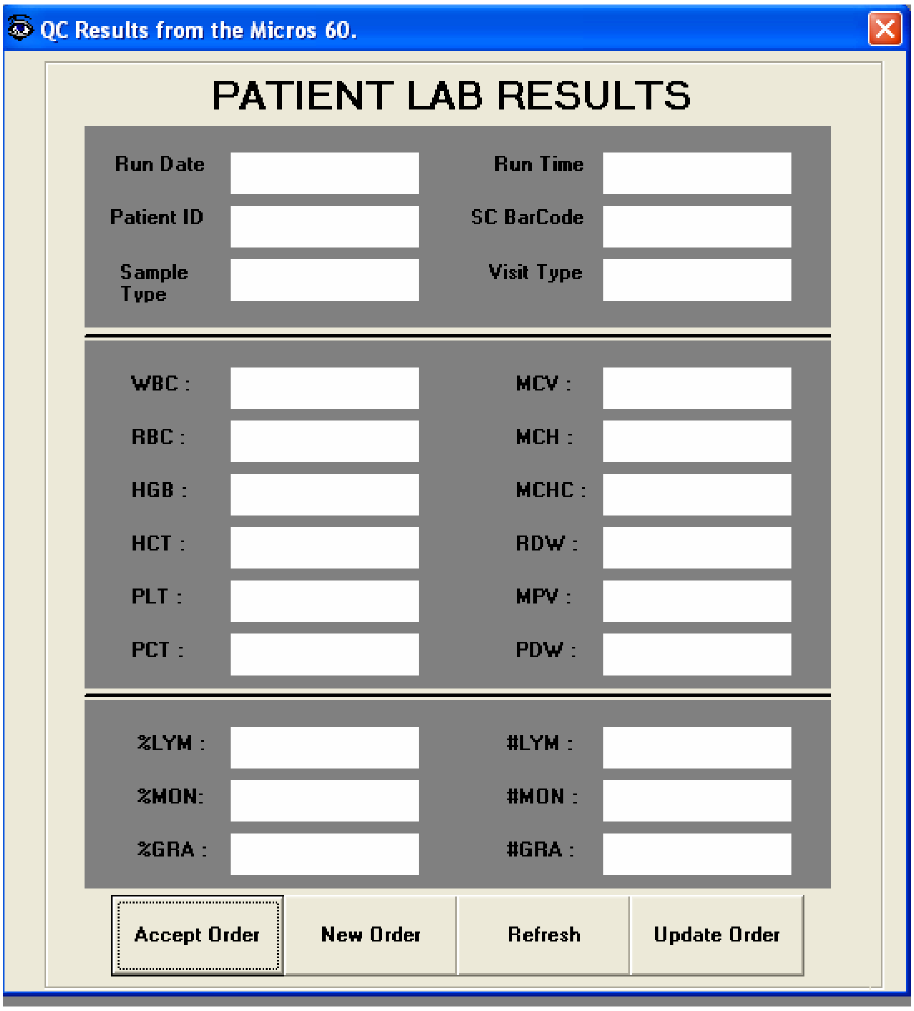


Figure 2.1 Form to capture the patient information at run-time

StudyID(PK)	Fname	LName	Age	Sex
02000221	Evan	Michael	32	M
02000121	Juletha	Kennedy	26	F
02000219	Shania	Twaine	37	F
02000012	Augulera	Tosh	24	F
02000021	Sharon	Jackson	21	M

PatientID(PK)	SCBarcode#(PK)
02000221	S200000704
02000121	S200000101
02000121	S20000245
020000121	S200000119
02000021	S200000102

SCBarcode#(PK)	SCBarcode#(PK)	VisitType	SampleType	WBC	---	User_ID(FK)
S200000204	S200000204	W2	PLC	003.3	---	N001
S200000101	S200000101	RC	PER	005.1	---	N007
S20000245	S20000245	RC	CD	004.9	---	N007
S200000121	S200000121	RC	PLC	004.2	---	N002
S200000102	S200000102	W2	PLC	004.7	---	N019

User_ID(PK)	FName	LName	Title
D001	Jonson	Peter	Doctor
N007	Amanda	John	Nurse Officer
N002	Devotha	Pius	Nurse Officer
N001	Shalon	Dennis	Nurse Officer
L019	Daniel	Paul	Lab Technician

Figure 2.2 Relational database schema for storing captured information

SUGGESTIONS AND RECOMMENDATION

The byte exchanged between PCs and smart devices may contain commands, data, or any other information. Even in the simple serial communication, it is important to ensure that all messages get to their intended destinations without errors, and that each node responds only to those messages intended for it. The method proposed is limited only to Haematology analyzer with ABX data formats. The similar study can be extended for exploring the ability to develop a general model to capture digital information from any medical care equipment with similar function as haematology analyzer. The study has focused only on Morogoro regional Hospital at Malaria Unit. In this case, the findings can be adopted by any organization dealing with malaria study with similar analyzer. However, some modification can be made so that the system can be integrated with any other healthcare systems.

CONCLUSION

The problems of storing data to XMLs formats have been solved by automatically loading patient results from Micros 60 OS/OT to PC database system at run-time. The COTRACOCO convention developed can be applied to extract digital information (zero and ones) from any digital devices and it has demonstrated that open source serial applications can be developed using this convention for the purpose of collecting characters composed from stream of bits (i.e. zeros and ones transmitted Micros 60 OS/OT machine).

Finally, any digital devices that use zeros and ones as means for communication can adopt the COTRACOCO technique. The COTRACOCO technique provides a new pathway for any programmer opting to develop the robust and reliable Laboratory Information Management Systems (LIMS) for healthcare centers and hospitals. This technique would not require the programmer to know the encoded data formats during serial port data exchange, just will be required to study the final result layout (i.e. final printed reports from the digital device) and hence extract the same data block from the Character Arrays composed during in the last stage of COTRACOCO convention.

REFERENCE

ABX Diagnostics, (2003), Micros 60 OS/OT User Manual. P/N: RAB042DA, Horiba Group, France.

Axelson, J., (2000), Serial Port Complete. Programming and Circuits for RS-485. Lakeview Research, USA.

Balena, F., (1999), Programming Microsoft Visual Basic 6.0. Microsoft Press, Redmond, Washington.

Coiera, E., (2004), Four Rules for the re-invention of health care. BMJ; Vol 328: pp 1197-1199.

Cooper, I. C., Crowther, A., Hughes, S., and Webb-Peploe, M.M., (1989), Microcomputer networking in Hospital, J R Soc Med; Vol 82: pp 285-87.

Masys, D.R., (1998), Advances in IT Implications for medical Education. West J Med; Vol 168: pp 341-347.

Ellis, B.W., Michie, H.R., Esufali St., Pyper R.J.D., and Dudley, H.A.F., (1987), Development of a microcomputer based system for surgical audit and patient administration, a review JR Soc Med; Vol 80:157-161.

Markivee, C. R., (1985), Networking of Microcomputers in radiology department, AJR Vol 145: pp 849-53.

Nelson, M.G., (1969), Automation in the Laboratory, J. Clic. Path 1969; 22:1-10.

Stock, S., Young, M., Hardman P.J., and Petty, A. H., (1985), A Microcomputer based system for surgical audit, Br J Clin Pact;Vol 39: pp 261-266.

Turner, E., and Bolton, J., (2001), Required steps for the validation of Laboratory Information Management systems, Qual. Assur;Vol 9 (3-4): pp 217-224.