E-Health Telecommunication Systems and Networks, 2014, 3, 24-32
Published Online September 2014 in SciRes. http://www.scirp.org/journal/etsn
http://dx.doi.org/10.4236/etsn.2014.33004

A Digital System for Managing HL7/CDA Electronic Medical Records Stored in iButtons®

Edgar Lugo, Roberto Muñoz, Carla C. Vilachá, Ángel Villegas, José Pacheco, Ricardo Villegas

School of Electrical Engineering, University of Carabobo, Valencia, Venezuela
Email: ealugo@uc.edu.ve, rcmunoz@uc.edu.ve, ccvilacha@gmail.com, avillegas@uc.edu.ve, jpacheco@uc.edu.ve, ravillegas@uc.edu.ve

Received 9 July 2014; revised 10 August 2014; accepted 27 August 2014

Abstract

The goal of this research was to develop a digital system that could allow managing electronic medical records (EMRs) codified under specifications of the Health Level 7/Clinical Document Architecture (HL7/CDA) international standard, and saving them in a portable digital storage device called iButton®. To this end, an USB-based hardware interface for reading and storing EMRs in iButtons was designed and implemented. In addition, a software application for invoking read/write operations on stored EMRs and showing their content on a graphical user interface was also developed, following the Extreme Programming (XP) software development methodology and using Visual Basic .NET as programming language. Tests conducted on the hardware interface showed that it was able to recognize any iButton type, reading and writing data on them as well. Moreover, the system helped in creating empty EMRs in conformance with the HL7/CDA standard, adding and viewing information, and updating it in the iButton. This system offers an easy way for managing and visualizing medical records codified in HL7/CDA, and allows patients to take their updated medical history with them everywhere.

Keywords

Medical Records, HL7/CDA, iButton

1. Introduction

Electronic Medical Records (EMRs) have been an important research field in medical informatics. According to Iakovidis I. [1], an EMR represents digitally stored medical information of a person's life, with a purpose of

How to cite this paper: Lugo, E., Muñoz, R., Vilachá, C.C., Villegas, Á., Pacheco, J. and Villegas, R. (2014) A Digital System for Managing HL7/CDA Electronic Medical Records Stored in iButtons®. *E-Health Telecommunication Systems and Networks*, **3**, 24-32. http://dx.doi.org/10.4236/etsn.2014.33004

supporting healthcare, education and research continuity, ensuring confidentiality of its contents at all times.

Nowadays, organizations providing healthcare services, store electronic medical records in several proprietary formats and manage them using commercially available medical information systems, such as EMR MedicWare, HealthFrame and MedFile, and among others, this situation has become a serious interoperability problem in the medical informatics field [2].

In this context, organizations, industries and researchers, including Health Level 7, European Committee for Standardization (CEN), IEEE, ANSI, ASTM and ACR-NEMA to just name a few, have been dedicated to developing standards in order to regulate digital storage of patients' medical information on a worldwide scale, thus allowing for interoperability between healthcare information systems.

This article presents a digital system for managing electronic medical records codified following the specifications of the HL7/CDA standard, which are then stored in portable electronic devices called iButtons, and given to patients so that they can take their updated medical history with them everywhere.

2. Resources

This section summarizes the most important resources used during the development of this research work

2.1. Health Level Seven-Clinical Document Architecture

Health Level Seven (HL7) is an organization accredited by the American National Standards Institute (ANSI), whose mission is to provide interoperability standards for integrating, managing and exchanging data between healthcare information systems that support clinical patient care [2].

HL7 Standard Version 2 was approved by ANSI in 2004 [3], and at present time is the most widely used for the exchange of clinical and administrative data between software applications [4]. However, because this version had drawbacks due to its excessive flexibility and lack of supporting information model, the HL7 Version 3 [5] was developed in order to solve them. This version is based on the Reference Information Model (RIM) [6] and proposes a standard tag-based document to represent electronic medical records named Clinical Document Architecture.

Clinical Document Architecture (CDA) Version 2.0 appears to meet the needs for sharing electronic medical records between systems in a standardized way, so in this context, HL7/CDA is a standard that specifies the structure and semantics of XML-based clinical documents [7].

2.2. iButton®

The iButton® is a small and portable device containing an integrated circuit inside a 16 mm thick stainless-steel button. It is durable enough to withstand extreme indoor or outdoor environments, so it could be used almost everywhere and on a daily-basis for applications such as access control and data logging [8]. **Figure 1** shows an image of the iButton.

The device uses its stainless steel exterior as a connection interface. It has an input/output contact and other for ground. Each contact is connected to the integrated circuit contained in the button and separated by a polypropylene ring that provides the necessary isolation for the device's operation. By using the aforementioned contacts, the device can establish communication through the 1-Wire® protocol, which supports two possible speeds: standard mode at 16 kbps, and overdrive mode at 142 kbps. Each iButton has an unchangeable serial number stored in the device's ROM, which could be used as an identification key because its uniqueness is guaranteed by its manufacturer, Dallas Semiconductor [8].

2.3. Visual Studio 2005 Express Edition IDE

Visual Studio 2005 Express Edition [9] is a freely available integrated development environment that supports several programming languages such as Visual Basic .NET (VB.NET), Visual C#, Visual C++ and Visual J#, and is used for deployment of software applications.

2.4. Wire® Software Development Kit (SDK)

The 1-Wire® Software Development Kit is a set of free libraries provided by Dallas Semiconductor, which allows to develop software applications for the Windows platform that require to communicate with hardware devices for sending and receiving information through the 1-Wire® protocol.

Currently, Dallas Semiconductor offers five library packages that can be selected depending on the target operating system and programming language: 1-Wire® Public Domain, 1-Wire® API for JAVA (OWAPI), 1-Wire® COM (OWCOM), 1-Wire® API.NET and 1-Wire® TMEX API [10]; the 1-Wire® API.NET is the one used in this work.

3. Methodology

In the planning phase, a questionnaire-type survey with eight questions was applied to several physicians in order to collect the information fields that should be contained in an electronic medical record, as well as their presentation order. These requirements were set down into user stories as indicated by the XP software development methodology [11], and functionalities derived from them or proposed by the authors were specified with an UML Use Case diagram [12], as shown in **Figure 2**.

After finishing specifying the user stories, it was gained a general overview of the application to develop, which was then used as a base for planning the software deliveries. For this purpose, an estimated completion time (in weeks) and a priority, with numeric valued ranged between 1 (one) and 4 (four), were assigned to each user story; the lower the value, the higher the priority. Planning of delivery was then performed by starting with user stories having higher priority until reaching the lower ones, resulting in a scheduled development time of 12 (twelve) weeks.

During the design phase, the software architecture was specified, choosing a 3-layer structure for it: Storage, Application Domain and Presentation. The Storage layer is represented by an HL7/CDA document stored into an iButton, which is sent through the hardware interface to the Application Domain layer. This layer has classes that process the information contained in the document and finally delivers it to the Presentation layer, which displays the information in the user interface running over a web browser. **Figure 3** shows these operating stages of the system.

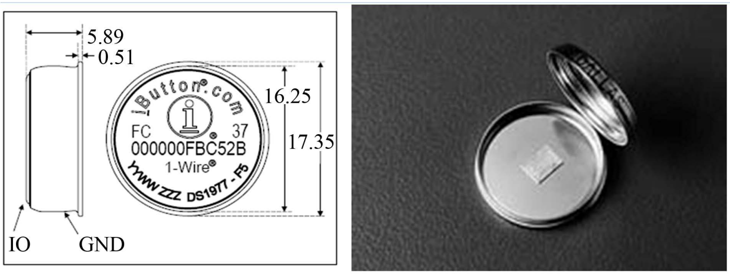


Figure 1. iButton® Source: [8].

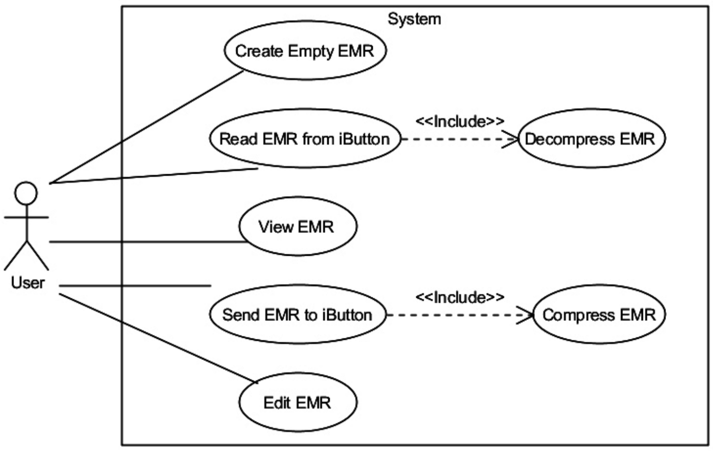


Figure 2. UML use case diagram for the EMR managing system.

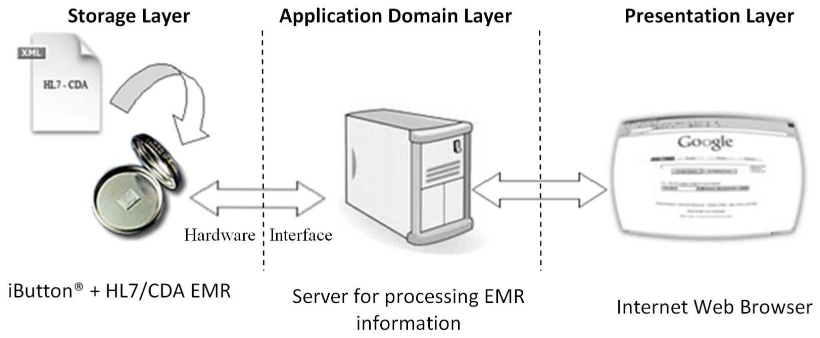


Figure 3. Operating stages for the EMR managing system.

After setting the system's operating scenario, a state diagram describing the desired behavior of the system was produced, as shown in **Figure 4**.

The programming phase began with a revision of the Clinical Document Architecture 2.0 in order to find out about the labels belonging to the standard, as well as their appearance order and meaning within a clinical document.

Understanding of each label was aided by using a file named cda.xsl, which is an XSLT (Extensible Stylesheet Language Transformations) conversion file [13] provided by HL7. This file allows transforming an HL7/CDA electronic medical record into an HTML (HyperText Markup Language) page [14] that could be displayed later on any web browser, while providing a better distinction of the elements contained in the medical records.

The implementation of the system's component for EMR edition was modeled following a method proposed by Qualls [15] for editing an XML file through a web browser with XSLT and ASP programming language. In this method, the information contained in the XML document is set within editable text fields, and changes are sent back to replace the original XML file after making proper modifications, as illustrated in **Figure 5**. A reengineer process was applied to transform the source ASP code into ASP.NET code, so allowing a subsequent integration of other functions and libraries needed for developing the system.

Modifications on the original cda.xsl file were also applied, producing a customized XSLT document named cdaEditable.xsl. Initially, only conversion to HTML of EMR's permanent or rarely changing data was carried out, e.g., medical record number, patient's name or history's creation date. This data has an exact location within the HL7/CDA EMR; therefore, it was specified as a short cut in XPath (XML Path Language) [16] in the cdaEditable.xsl file. Remaining information, conformed by sections whose data could change, e.g., type of EMR, medical institution creating the EMR, etc., was translated to HTML afterwards, performing a full scan of the EMR to extract required information while keeping its original data format. Data displaying in edition mode was made through editable text fields, each one having a related hidden text field containing the document location (in XPath) from where it was extracted, so that it could be easier to update changes made by users.

The next step was to select an iButton suitable to store, read and update EMRs with an average size of 30 Kb. This size was estimated from a set of HL7/CDA documents available for educational use in the Ringholm online repository [17]. Features considered for this selection were type and size of memory, recommended application area, cost and lifespan; **Table 1** lists these features for several commercially available iButtons.

After evaluating several commercially available iButtons, the DS1977 model was selected because it allows read/write operations, costs less than the NVRAM model, its lifespan is ideal for preserving the clinical history of a patient for a long period of time, and according to its manufacturer, the device is suitable as storage medium for data coming from inspections, maintenances, audits and medical procedures, among other sources [18]. The DS1977 provides a 32 Kb memory, structured into 509 pages of 64 bytes each, which meets the storage capacity requirements of the system. Furthermore, access to the device's memory can be controlled through two different passwords to enable either read-only or read/write modes, respectively.

Designing and building a hardware interface to exchange information between a personal computer (PC) and the previously selected iButton was the subsequent step in the implementation phase. This interface uses the 1-Wire® protocol to read the information stored in the DS1977 and send it to the PC, and the other way around. For constructing the interface, two main electronic components were required:

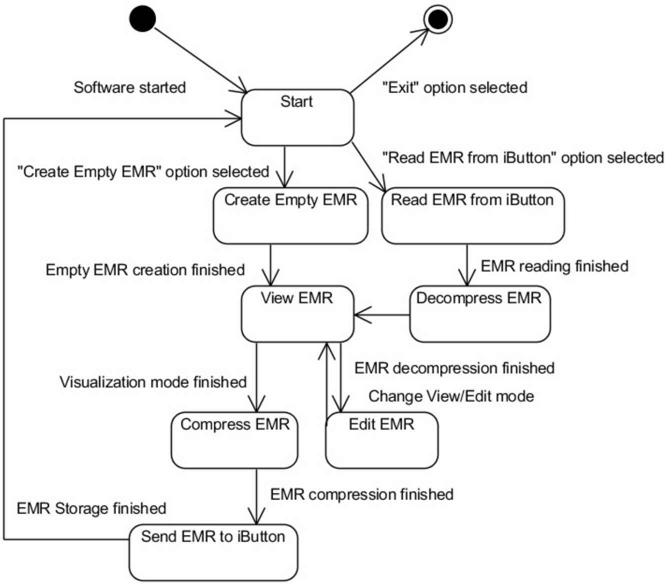
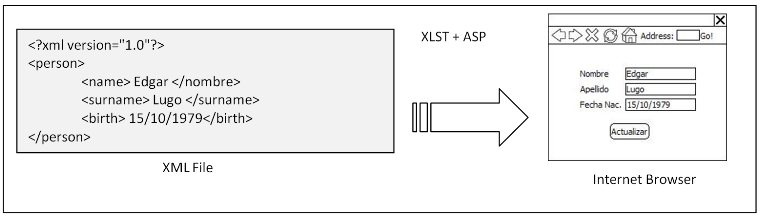


Figure 4. State diagram for the EMR managing system.


```
<?xml version="1.0"?>
<person>
        <name> Edgar </nombre>
        <surname> Lugo </surname>
        <birth> 15/10/1979</birth>
</person>
```

Nombre Edgar
Apellido Lugo
Fecha Nac. 15/10/1979

Figure 5. Diagram of a method for editing an XML file with XSLT and ASP.

Table 1. Commercially available iButton® models.

Model	Memory Type		Memory Size	Application	Cost	Lifespan
DS1992L	NVRAM	Allow operations in environments where electrical contacts can be intermittent	1 KB	Applications that require data updating on a high frequency	High	Less than 10 years
DS1993L			4 KB			
DS1995L			16 KB			
DS1996L			64 KB			
DS1982	EPROM	Allow writing into the iButton's memory until reaching its full capacity	1 KB	Applications that handle permanent data but require appending information to it	Medium	Less than 10 years
DS1985			16 KB			
DS1986			64 KB			
DS1971	EEPROM	Allow both read and write operations	256 Bits	Applications that require data updating on a moderate frequency	Low	Greater than 10 years
DS1972			1024 Bits			
DS1973			4 KB			
DS1977			32 KB			

- **FT232RL:** integrated circuit responsible for USB (Universal Serial Bus) to serial (with TTL compatible voltage levels) conversion. It comes in a 28-terminals SSOP package [19], which can be manually installed without special techniques or welding machines. For working with a PC, this component requires a driver available for different operating systems that can be downloaded from the Future Technology Devices International Ltd. website (www.ftdichip.com).

- **DS2480B:** integrated circuit that converts from serial port to 1-Wire bus, allowing any master (host) with UART serial communication to invoke operations under the 1-Wire® protocol and return results back to the host, as shown in the simplified functional diagram of **Figure 6**.

In this design, an integrated USB port enables the communication between the PC and the FT232RL component, which in turn is connected to the DS2480B integrated circuit. The interface, showed in **Figure 7**, is also composed of a support for inserting a single iButton, a bicolor LED for indicating the device's operative status and a B-type USB female connector for enabling a connection with any standard USB printer cable. In addition, a standard 0.1" connector is located at the device's bottom in order to allow connecting other 1-Wire devices and in doing so, to take advantage of the intrinsic networking capacity of this technology.

Drivers and configuration software, provided by Dallas Semiconductor for 1-Wire on Windows platforms, were later downloaded and installed; these applications allowed selecting a COM port and enabling it for accessing the hardware interface. A test program was used to detect and identify an iButton inserted into the interface; it produced as result an immediate recognition of the device (see **Figure 8**), thus validating the correct operation of the built interface.

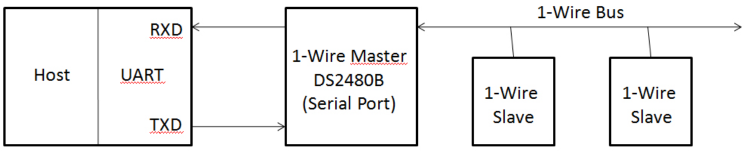


Figure 6. DS2480B simplified functional diagram.

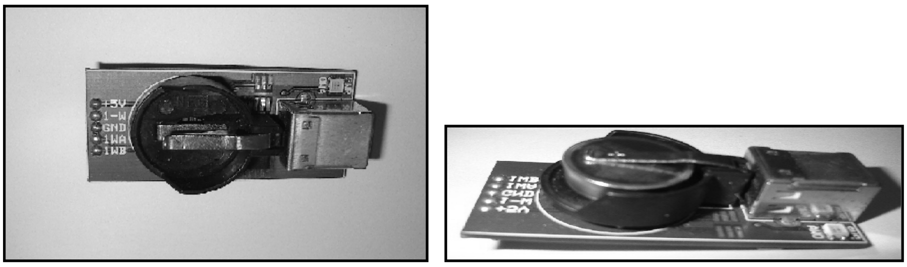
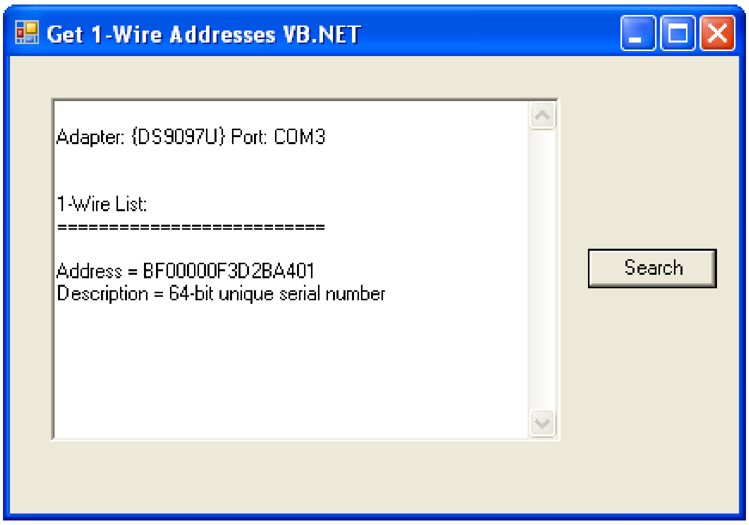


Figure 7. Hardware interface for exchanging information between a PC and the iButton.



```
Adapter: {DS9097U} Port: COM3

1-Wire List:
==========================

Address = BF00000F3D2BA401
Description = 64-bit unique serial number
```

Figure 8. Results of identifying an iButton with the testing software.

A library, named iButtonCom, was developed to integrate the iButton's interface-based data read/write functionality into the EMR management module. The library is based on a sample project also provided by Dallas Semiconductor, and contains the methods described bellow:

- **ReceiveFile**, requests access to the 1-Wire network, sets the transmission speed of the hardware interface and then **proceeds** to read the EMR's size (in bytes) and the number of pages required by the EMR, from first page of the iButton's memory. After retrieving these values, the method traverses all the EMR's pages while reading blocks of 62 bytes from each one, in order to fill a temporary array with their information. Next, this array is stored into a file on the PC and the network connection is closed, releasing the communication port afterwards; the file will be used by the EMR management module for editing and updating the record.
- **SendFile**, takes a file containing an updated HL7/CDA EMR as input argument, and then converts it into a temporary array of bytes. Next, it requests access to the 1-Wire network, sets the transmission speed and requests the iButton's **page** capacity from the hardware interface; this capacity will be used to check whether or not the device can hold the EMR's entire content. Having passed the capacity checking process, the EMR's size and its required number of pages are written at first page of the iButton's memory, and the temporary array's content is written in blocks of 62 bytes on each remaining page, until completing the full storage process. After that, the network connection is closed and its related communication port released.

As a final step, a compression library named UtilCompresion was also developed in order to improve the storage capacity of the iButtons. This library makes use of compression functions based on the GZip format [20] and available within System.IO.Compression namespace of .NET Framework 2.0 [21]. UtilCompression has methods for compressing and decompressing an EMR, both of which were integrated into SendFile and ReceiveFile methods of iButtonCom library, respectively.

4. Results

The EMR management system was tested on a Windows XP platform, and its performance evaluated on different aspects. In the medical data management aspect, the system allowed to create empty medical records and later adding, updating or displaying its information in conformance with HL7/CDA standard. Using an XSLT transformation template made it possible setting or changing properties of clinical history's elements without modifying the software's source code, thus giving healthcare institutions enough flexibility for applying custom XSLT files that satisfy their specific requirements for edition and visualization of medical records.

Regarding the hardware interface developed, tests showed that the device was able to recognize any type of iButton and execute read/write operations on them, achieving a better velocity performance on these operations than adapters sold by Dallas Semiconductor.

Tests were also conducted for evaluating the performance of the iButtonCom library in successfully exchanging electronic medical records to and from the hardware interface. Registered times, using files of 4, 8 and 16 Kb, vary between 13 and 54 seconds for data reading processes, and between 7 and 20 seconds for data writing processes, as shown in **Figure 9**.

The UtilCompression library allowed compressing and decompressing EMRs with an average compression rate of 85%. Results of compression tests are shown in **Figure 10**, where it can be noticed that the largest size of a file that can be stored in an iButton using this compression algorithm is 1500 Kb.

5. Conclusions

Starting from a survey applied to several physicians, it was possible to obtain information for creating an electronic medical record template, in conformance to HL7/CDA specifications, which can hold enough information of a patient to satisfy the requirements of physicians and healthcare centers.

The iButton®, a device selected as storage media for the system, has features that make it attractive and safe for storage of medical records, such as high durability, lifespan of at least ten years and data integrity protection through access passwords. Regarding the hardware interface built for exchanging information between the system and the storage media, it proved to be able of recognizing any commercially available iButton, and its convenient size and ease of use make from it an adapter device that could be feasibly implemented in any healthcare facility.

Finally, results obtained from performance and compression tests show that libraries iButtonCom and Util

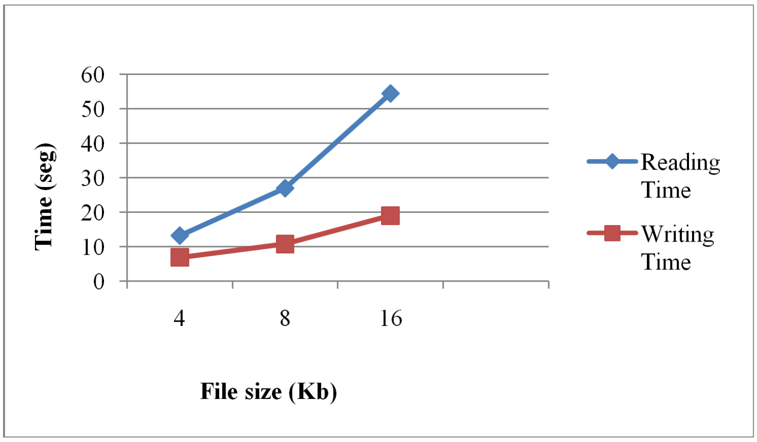


Figure 9. Results of performance tests for iButtonCom library.

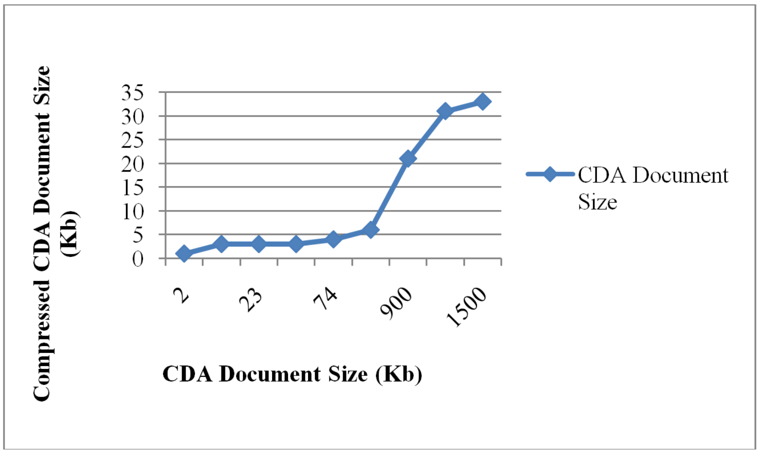


Figure 10. Results of compression tests for UtilCompression library.

Compression provide good data transfer times to operations of reading and writing in the storage media, as well as a compression rate that could improve the storage media capacity. Moreover, the system's software allows creating, updating and displaying HL7/CDA electronic medical records in a simple and easy way, hiding low-level complexities of managing and storing them from users.

References

[1] Iakovidis, I. (1998) Towards Personal Health Records: Current Situation, Obstacles and Trends in Implementation of Electronic Healthcare Records in Europe. *International Journal of Medical Informatics*, **52**, 105-117. http://dx.doi.org/10.1016/S1386-5056(98)00129-4

[2] Health Level Seven (2014) About HL7. http://www.hl7.org/about/index.cfm?ref=nav

[3] Health Level Seven (2006) HL7® Version 2 Standard. http://www.hl7.com.au/V2-Resources.htm

[4] Eichelberg, M., Aden, T. and Riesmeier, J. (2005) A Survey and Analysis of Electronic Healthcare Record Standards. *ACM Computing Surveys*, **37**, 277-315. http://dx.doi.org/10.1145/1118890.1118891

[5] Health Level Seven (2006) HL7® Version 2 Standard. http://www.hl7.org.au/HL7-V3-Resrcs.htm

[6] Health Level Seven (2006) HL7 Reference Information Model 2.14. http://www.hl7.org/library/data-model/RIM/modelpage_mem.htm

[7] Health Level Seven (2005) HL7 Clinical Document Architecture, Release 2.0. 190 p.

[8] Dallas Semiconductor MAXIM (2008) What Is an iButton? http://www.maxim-ic.com/products/ibutton/ibuttons/

[9] Microsoft Corporation (2005) Visual Studio Express Editions.

http://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx

[10] Dallas Semiconductor MAXIM (2005) 1-Wire Software Developer's Kit (SDK) for Windows. http://www.maxim-ic.com/products/ibutton/software/windowsdk/

[11] Beck, K. (1999) Extreme Programming Explained: Embrace Change. Addison-Wesley Professional, Estados Unidos.

[12] Object Management Group (2008) UML® Resource Page. http://uml.org/

[13] World Wide Web Consortium (1999) XSL Transformations (XSLT) Version 1.0. http://www.w3.org/TR/xslt/

[14] World Wide Web Consortium (1999) HTML 4.01 Specification. http://www.w3.org/TR/html4/

[15] Qualls, M. (2007) Editing XML with XSL and ASP. http://www.xmlfiles.com/articles/michael/editingxml/default.asp

[16] World Wide Web Consortium (1999) XML Path Language. http://www.w3.org/TR/xpath

[17] http://www.ringholm.de/download/CDA_R2_examples.zip

[18] Dallas Semiconductor MAXIM. DS1977 Password-Protected 32Kb EEPROM iButton. http://datasheets.maxim-ic.com/en/ds/DS1977.pdf

[19] Future Technology Devices International Ltd. (2007) FT232 USB UART I.C. Datasheet. http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf

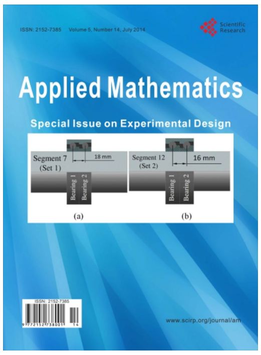
[20] Internet Engineering Task Force (1996) DEFLATE Compressed Data Format Specification. http://www.ietf.org/rfc/rfc1951.txt?number=1951

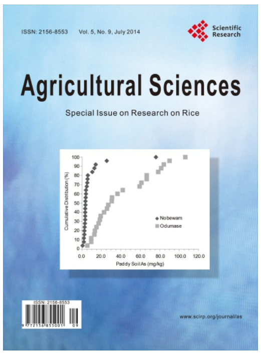
[21] Microsoft Corporation (2005) System.IO.Compression Namespace. http://msdn.microsoft.com/en-us/library/system.io.compression.aspx

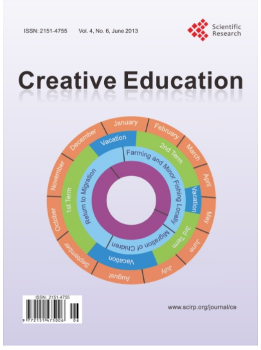
Advances in
Bioscience and Biotechnology

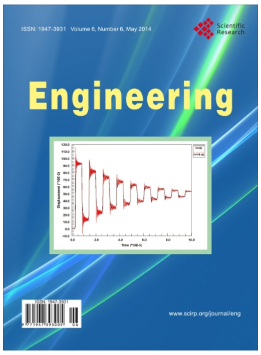
American Journal of
Analytical Chemistry

American Journal of
Plant Sciences
Special Issue on Photosynthesis

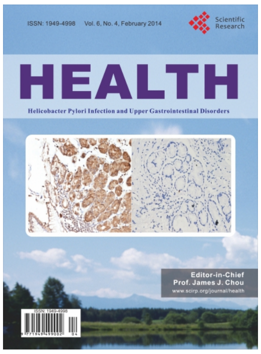

Applied Mathematics
Special Issue on Experimental Design


Agricultural Sciences
Special Issue on Research on Rice


Creative Education

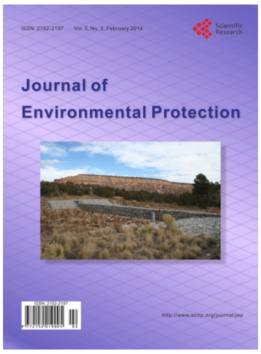

Engineering

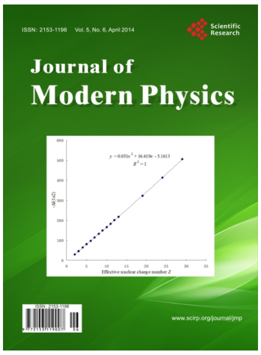
Food and Nutrition Sciences


HEALTH
Helicobacter Pylori Infection and Upper Gastrointestinal Disorders
Editor-in-Chief
Prof. James J. Chou

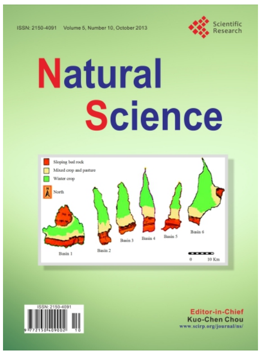
Journal of Computer
and Communications

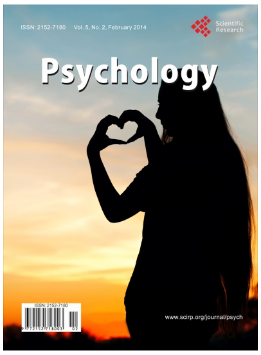
Journal of Cancer Therapy
Special Issue on Bone Tumor Research


Journal of
Environmental Protection


Journal of
Modern Physics

Modern Economy


Natural
Science
Editor-in-Chief
Kuo-Chen Chou


Psychology