

Reflections on the Role of Open Source in Health Information System Interoperability

S. Sfakianakis¹, C.E. Chronaki¹, F. Chiarugi¹, F. Conforti², D.G. Katehakis¹

¹Institute of Computer Science, Foundation for Research and Technology—Hellas, Heraklion, Crete, Greece

²Institute of Clinical Physiology, National Research Council, Pisa, Italy

Summary

Objectives: This paper reflects on the role of open source in health information system interoperability. Open source is a driving force in computer science research and the development of information systems. It facilitates the sharing of information and ideas, enables evolutionary development and open collaborative testing of code, and broadens the adoption of interoperability standards. In health care, information systems have been developed largely ad hoc following proprietary specifications and customized design. However, the wide deployment of integrated services such as Electronic Health Records (EHRs) over regional health information networks (RHINs) relies on interoperability of the underlying information systems and medical devices.

Methods: This reflection is built on the experiences of the PICNIC project that developed shared software infrastructure components in open source for RHINs and the OpenECG network that offers open source components to lower the implementation cost of interoperability standards such as SCP-ECG, in electrocardiography.

Results: Open source components implementing standards and a community providing feedback from real-world use are key enablers of health care information system interoperability.

Conclusions: Investing in open source is investing in interoperability and a vital aspect of a long term strategy towards comprehensive health services and clinical research.

Keywords

Open source, interoperability, standards, health information systems, electrocardiography

Geissbuhler A, Haux R, Kulikowski C, editors. IMIA Yearbook of Medical Informatics 2007. *Methods Inf Med* 2007; 46 Suppl 1: 51-61

1 Introduction

Open source is a philosophy, a development model, but above all a license model. According to the Berne convention without a software license, a program cannot be copied or modified without the explicit permission of the authors [1]. Most software licenses are proprietary. Proprietary licences do not allow users to access the source code of the program. Open source licenses, on the other hand, allow users to study, copy, modify, and in some cases release the modified program for the benefit of the community. They may be copyleft or non-copyleft. A non-copyleft license is a non-persistent open source license. A non-copylefted program may be redistributed, and users are allowed to make proprietary versions of the program. The BSD license [2], a non-copyleft licence, allows incorporating the open source software (OSS) in proprietary commercial products to be released under different licenses. That is particularly important for interoperability as it supports the consistent implementation of open standards and the easy integration of heterogeneous information systems. In contrast, a copyleft license is persistent. Redistribution of a copylefted program, either in modified or unmodified form, gives the recipients the same rights as the original program. A prerequisite for this is that the source code is distributed with the program. The most widespread copyleft license is the GNU

General Public License (GPL) [3]. GNU GPL requires copies and derivatives of the source code to be made available on terms no more restrictive than those of the original license.

Computers exchange health data in encoded formats. Software needs to implement correctly the data formats, otherwise massive data corruption may occur. Customers that save their data in the format of a particular vendor find themselves unable to choose a different vendor who does not implement that format even though the corresponding software may be superior (vendor lock-out). In healthcare, it is not perceived as sustainable to allow a single vendor to dominate the market and the wide use of open standards is a requirement for interoperability and continuity of care. The International Telecommunication Union defines open standards as “standards made available to the general public that are developed, approved, and maintained via a collaborative and consensus-driven process.” [4] Thus, a key element of open standards is a collaborative process that supports their voluntary and market-driven development following a transparent consensus-driven process that is reasonably open to all interested parties and is reasonably balanced to ensure that it is not dominated by any one interest group. Intellectual property rights to implement the open standard are either for free or on non-discriminatory terms that may include reasonable monetary compensation. Implementation of stan-

dards should be free of economical, political, or legal implications on their development and use. Open standards should facilitate interoperability among products and services, and their integration in realistic processes. Emphasis on quality and sufficient level of detail should permit the development of a variety of competing implementations of interoperable products or services. Standardized interfaces should be public and maintained by the Standard Developing Organization (SDO) responsible for the standard.

Specifically in health care, open source is linked interoperability which is a patient safety issue, as it minimizes error-prone processes and suboptimal care within and across health organizations world wide [5]. Interoperability, the ability of two or more systems or components to exchange information (functional interoperability) and to use the information (semantic interoperability) that has been exchanged [6], is a prerequisite to the establishment of an active life-long Electronic Health Record (EHR). Open standards are important when heterogeneous systems need to cooperate to support clinical workflows and exchange information to facilitate availability of patient information when and where needed, to reduce medical errors, and to protect patient safety. Interoperability can undoubtedly only be achieved with open standards and an OSS community that enables their consistent implementation and widespread adoption.

Standards in health care are “de jure” (in principle) and “de facto” (in practice). Besides the ISO TC215 [7], CEN TC251 [8] in Europe and ANSI [9] in the USA, which are the formal SDOs developing “de jure” standards, there exist independent organizations or professional societies supporting the development of eventually accredited “de facto” standards such as DICOM [10]

(led by the American College of Radiologists (ACR) and the National Electrical Manufacturers Association (NEMA)) and SNOMED [11] (led by the College of American Pathologists). Note that open standards doesn't always mean cost-free standards. In some cases like SNOMED, agreements are formed at the country level to maximise standardization benefits and manage the costs involved.

With Regional Health Information Networks (RHINs), the pressing need for wide expertise and multi-stakeholder consensus in addressing heterogeneous data exchange and complex workflows, became evident. Industry consortia like OMG CORBAMED [12] and independent organizations like HL7 [13] develop “de facto” standards. Finally, the need to collaboratively validate standards on the basis of integration profiles is addressed by IHE [14]. IHE integration profiles promote the coordinated implementation of standards in the context of concrete integration scenarios, supplementing the vendors' statements of conformance to standards such as DICOM and HL7. Validation exercises or Connectathons are carried out in HIMMS [15] and IHE meetings every year, testing new integration profiles and identifying pitfalls and possible limitations of existing standards.

Meanwhile, the development of a robust middleware infrastructure to support IHE-type integration scenarios became of paramount importance for RHINs aiming to support registries such as those for immunization and disease surveillance, integrated to health information systems in daily use [16]. Starting from OMG CORBAMED component specifications, a number of common components in open source were developed. In particular, common infrastructure components for security and confidentiality, resource identification & access (including patients,

health professionals, and healthcare facilities), as well as clinical data access through open interfaces, are considered necessary to establish regional services and enable collaboration among health care facilities. Using open standard interfaces and open source technologies to implement such components as well as creating an open source community to support the evolution, customization, and maintenance of these components was a key objective for the PICNIC project [17]. The PICNIC experience with the development of OSS components is presented in the next section. With electrocardiographs and other low cost medical devices the situation was different. The best that patients and carers could hope for is to get the patient's old ECG traces in paper or digitized. Despite the existence of SCP-ECG [18], an official interoperability standard in Europe (EN1064:2005, EN1064:2007), most digital electrocardiographs implement proprietary data formats, as OpenECG pointed out [19]. Section 3 reports on ECG interoperability standards, conformance testing, and experiences from the OpenECG open source repository arguing that OSS, converters, and conformance testing services are vital for the seamless integration of digital ECGs in the EHR and their effective use in eHealth services. Then, section 4 reflects on the role of open source in interoperability of health information systems, whereas section 5 presents the main conclusions.

2 PICNIC: Open Source Components for RHINs

PICNIC stands for Professionals and Citizens Network for Integrated Care and was a European Commission co-funded research and development

project, established under the 5th Framework of European Research “Information Societies Technology Programme”. PICNIC designed, developed, validated, and delivered a number of certified OSS infrastructure components for eHealth, in close co-operation with the industry [20]. Having eighteen RHIN services as a starting point, PICNIC through a carefully planned storyboard articulated the future healthcare response with the empowered citizen at the centre. Professionals provide services and support through a RHIN that aims to maintain the citizens’ quality of life despite clinical illness which impacts not only their family life, but also their ability to sustain an occupation. The initial set of RHIN services had been divided into three groups:

- **Clinical Services and Telemedicine** that involved *Clinical Messages, Clinical e-Mail, Clinical Booking, Shared Records, Care Protocols, Mobile and Emergency, Home-care Monitoring, and Telemedicine*
- **Health Information Services** that

involved *Surveillance Information, Yellow Pages, Professional Guidelines, Disease Quality Management, Public Health Information, and Continuing Professional Development*

- **Administrative Services and Electronic Commerce** that involved *Reimbursement, Electronic Commerce, Patient Identification (ID), and Resource Management*.

A common goal for all three groups has been the provision of reliable communication, sharing and access to one’s EHR [21] to enable safe and shared clinical care by means of an enabling Health Information Infrastructure (HII). Component development concentrated in three groups:

- **Clinical Messaging** consisting of highly structured patient-related information
- **Access to Patient Data** by means of accessing common information services
- **Collaboration** through immediate access to specialized expertise.

Sharing of know-how as well as source

code was necessary for PICNIC to deliver harmonized interoperable services throughout Europe by means of common enabling OSS components. All specifications relevant to the PICNIC architecture and associated standards, scenarios and models for patient-centred care, and common components development were made available in the public domain. Common components developed by PICNIC were released in open source, under the three clause BSD licence [20].

2.1 PICNIC OSS Components

An integrated EHR requires an enabling infrastructure that provides access to heterogeneous health information systems under appropriate security constraints. The relevant common components were delivered as follows (Figure 1):

- Clinical Observation Access Service (COAS) for obtaining patient data directly from Health Informa-

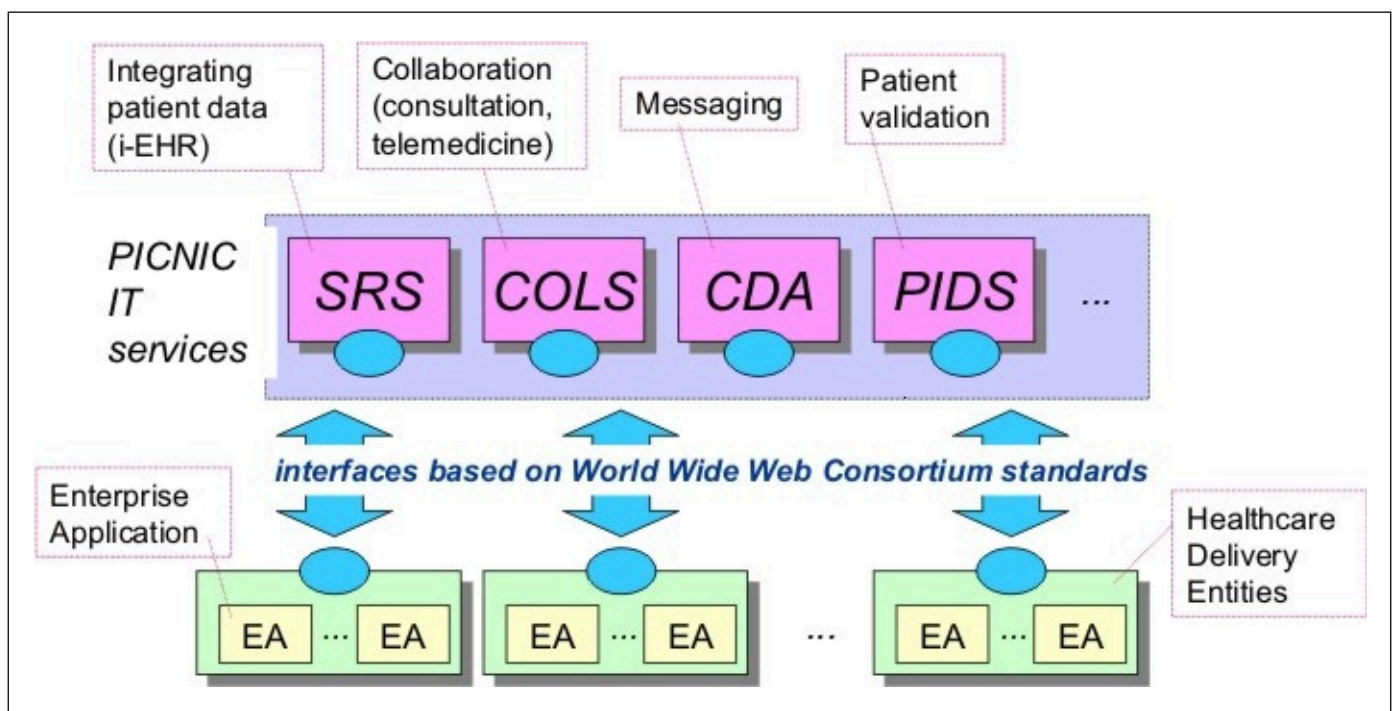


Fig 1 PICNIC IT services (PICNIC architecture)

tion Systems and updating patient information based on OMG COAS interface specification [22]

- Patient Identification Service (PIDS) for enabling unique patient identification across a RHIN and for the provision of a master patient index based on OMG PIDS interface specification [23]
- Shared Records Indexing Service (SRIS) for the management and indexing of meta-information on the content of the patient data in health information systems throughout the RHIN
- Shared Records Update Broker (SRUB) serving as a mediation tool for maintaining EHR consistency across the RHIN by propagating to SRIS modifications pertaining to patient data.

The collaboration component group delivered the following OSS components:

- Collaboration Service (COLS) for the provision of the platform that allows general practitioners and medical experts to share patient-related information in the context of a teleconsultation episode both inside and across RHINs
- Resource Service (RESS) for identification of healthcare related agents (e.g. organizations, devices, software, etc) and the means for accessing them and determining their availability.

For the development of these components a large number of OSS tools, libraries, compilers, etc. were used mainly originating GNU. Development support was provided by OSS mailing lists and newsgroups. This kind of support is certainly “unpredictable”, since there is no guarantee that the problem encountered will be answered soon if at all. Nevertheless, PICNIC found the OSS community to be highly responsive and very helpful while developing the components. The majority of issues faced in PICNIC proved to be minor problems that other OSS developers had already encountered. In addition, edu-

cational resources were deemed necessary to extend the group knowledge of TAO’s intrinsics [24]. Finally, the creation of the *PICNIC Open Community* was encouraged to establish a community of OSS developers who were committed to further development, enhancement, and re-use of the common PICNIC components.

2.2 PICNIC Results

PICNIC developed a new architectural paradigm for healthcare ICT based on an information infrastructure comprising a set of OSS components that support the architecture, a proactive assessment model, multiple case studies with independent evaluation reports and a certification model.

PICNIC used many existing OSS components, enhanced or extended some of these, and developed others. The resulting prototype subsystems were integrated successfully to pre-existing health information systems. However, undertaking socio-economic assessments of individual components, when they were embedded in an application or service comprised of many components, turned out to be a complex task. It turned out that European healthcare ICT service suppliers will not speculatively adopt an architectural framework for standards, no matter how robust the case for its use, without a clear demand for such a framework coming from the buyers of ICT services. Thus, the lead in adopting a RHIN architecture comes from the ICT policymakers in the regional health authorities. Many authorities, however, do not appreciate the need for an inter-enterprise integration architecture, as they are still in the business of trying to integrate islands of information within their organisations following “traditional” enterprise approaches. Thus, the need for inter-en-

terprise integration tools/ methods only becomes apparent at a later stage of RHIN development, when integration across organisational/ regional boundaries is attempted.

Therefore, PICNIC customers were members of the healthcare ICT vendor community who were helping to build and operate RHINs (i.e. the policy makers and technology procurement experts). PICNIC suppliers were a community of OSS component developers and health telematics researchers with shared interests and objectives.

The business case for using the PICNIC architecture and components was based on the economies of scale of using standards and components which are accepted across Europe and enable healthcare ICT vendors to develop software incorporating components that operate in any EC country/ region and support both the interoperability of products and their integration with existing legacy systems. Expected savings arise from adopting common standards and sharing the costs of software development.

3 OpenECG: Open Source for ECG Interoperability

Electrocardiography is the most frequently applied non-invasive examination for early detection and follow-up of heart disease, a leading cause of morbidity and mortality in western countries. Electrocardiograms (ECGs) should be exchanged between health care providers and stored in EHRs in a device-independent standard ECG format, to support continuity of care. Serial comparison of ECGs in the context of an individual’s EHR can provide valuable information and early warning on the progress of heart disease. However, due to inconsistent data formats and incompatible protocols, ECGs

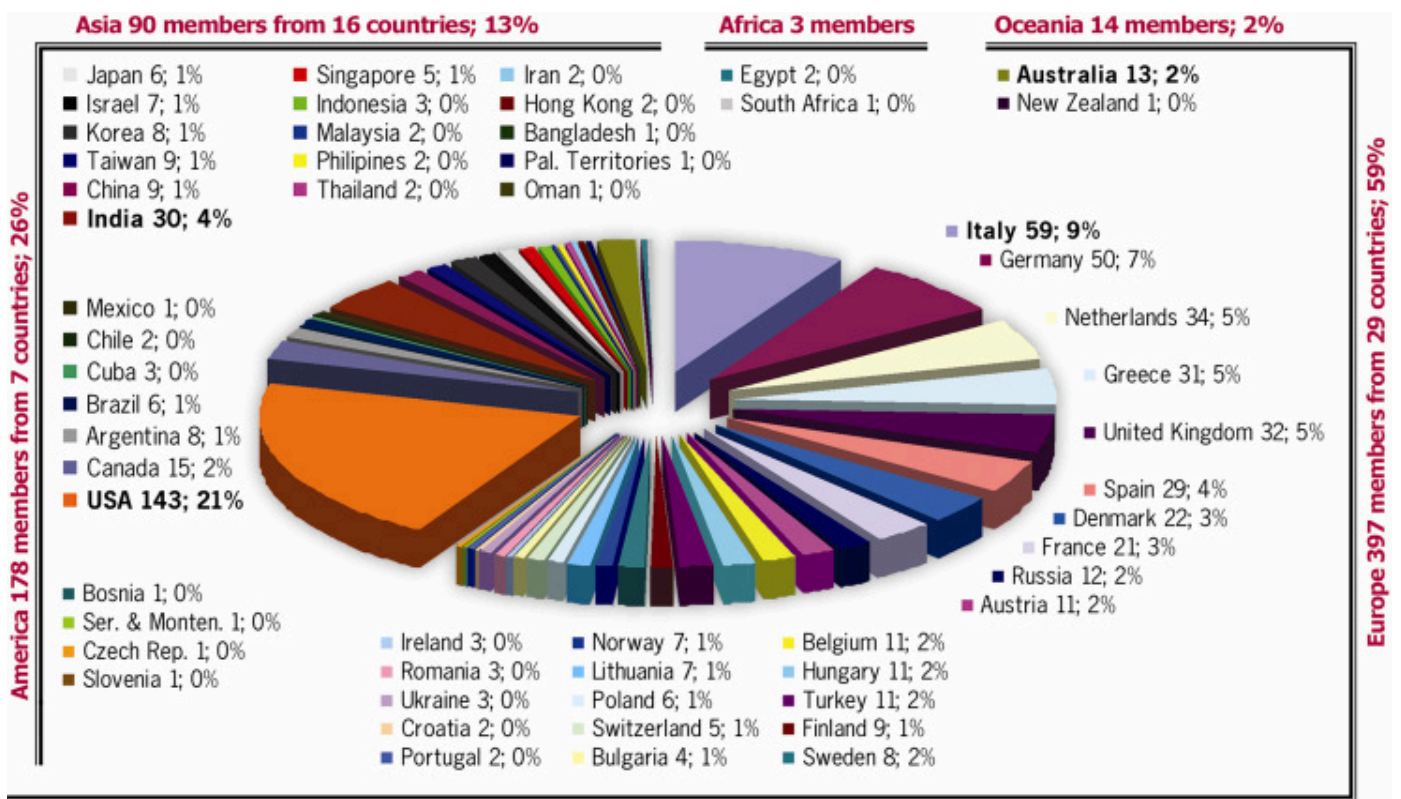


Fig. 2 OpenECG network in January 2006. At the end of 2006 there are more than 700 members from 58 countries worldwide

cannot be exchanged in a way that allows visualization in high precision, comparison with previous ECGs, and interpretation in the EHR context. This problem led to the creation of the OpenECG network in 2002 (see Figure 2) to encourage and promote the use of ECG interoperability standards, to consolidate expertise and best practice in ECGs interoperability standards, to provide assistance in implementation with open source tools & data sets, and to offer interoperability testing services.

3.1 ECG Interoperability Standards

SCP-ECG is the European standard [18] for ECG-specific data exchange to achieve interoperability between different ECG devices and eHealth systems. SCP-ECG is a standard communication protocol that specifies the interchange format and a messaging

procedure for ECG device-to-computer communication and uploading ECG records from the computer to the ECG device. Since March 2005, SCP-ECG (EN1064:2005) is the European standard for high quality diagnostic ECG exchange, and if consistently implemented, it ensures interoperability. An overview of the SCP-ECG data format appears in Figure 3.

Beyond SCP-ECG there are several open and numerous proprietary ECG standards. The most well-known ones are the HL7v3 Annotated ECG format [25] created to support full disclosure in clinical trials and DICOM Supplement 30 [26]. Both of them are appropriate for the storage of waveform data. There is also MFER, a Japanese standard for the storage of waveforms including ECGs [27]. Moreover, the IMEX project is developing MSD [28], a new micro-system data format (based

on SCP-ECG section 1), to meet the requirements of wearable devices. Research centers use mainly raw data originating from laboratory instruments, which typically generate interleaved channel data. If signals have to be exchanged a simple signal-oriented format like EDF [29] or the MIT format [30] is typically used. Optimizations of data space and transmission time, as well as diagnostic quality, which are issues of primary relevance to the SCP-ECG standard, are not of major concern.

Nevertheless, it is rather difficult for integrators to implement SCP-ECG correctly and there are variations in implementations, which can be a barrier to interoperability. Several ECG device manufacturers and integrators have implemented the SCP-ECG standard, but that alone does not attain the interoperability objective. This is due

Status	Content
Mandatory	2 bytes- Checksum CRC - CCITT over the entire record
Mandatory	Pointers to data areas in the record [0]
Mandatory	Header Information – Patient data/ECG Acquisition data [1]
Optional	Huffman tables used in encoding of ECG data [2]
Optional	ECG lead definition [3]
Optional	QRS Locations (if reference bits are encoded) [4]
Optional	Encoded reference beat data if reference beats are stored [5]
Optional	Residual signal after reference beat subtraction or encoded rhythm data [6]
Optional	Global measurements [7]
Optional	Textual diagnosis from the "interpretive device" [8]
Optional	Manufacturer specific diagnostic and over reading data.. [9]
Optional	Lead measurement results [10]
Optional	Universal statement codes resulting from the interpretation [11]

Fig. 3 The SCP-ECG data structure

partly to misconceptions and partly to the lack of widely publicised conformance statements that exist for other standards such as DICOM. Secondly, although ECG viewers and converters exist for alternative ECG formats, terminology issues are not adequately resolved. Thirdly, achieving interoperability is a very costly procedure, since a non-disclosure agreement is still required to integrate a closed ECG device. OpenECG advocates that data format and protocol specifications, open source tools, and certification services should be freely available to reduce the cost of device integration. For that purpose, OpenECG launched the first

online certification service and the OpenECG open source repository.

3.2 OpenECG Certification and Helpdesk

An SCP file may include demographics, ECG channels, measurements, and diagnostic statements as dictated by the conformance levels of SCP-ECG (Figure 4). In 2003, OpenECG established an on-line conformance testing service to support the OpenECG community at large in implementing interoperable eHealth systems with SCP-ECG support. A member may submit an ECG file in an alleged SCP-ECG format and

Category	Data Sections Required	Content Description
I	0, 1, 7,8	Demographics, global measurements, and interpretation
II	0,1,2,3,6, (7), (8)	Demographics and ECG rhythm data
III	0,1,2,3,5, (7), (8)	Demographics and reference beats
IV	0,1,2,3,4,5,6, (7), (8)	Demographics, rhythm data, and reference beats

Fig 4 Conformance types for the SCP-ECG standard

receive a list of errors and warnings. If no errors are detected, the submitter may request a certificate that is granted after a more thorough manual review of the file. The conformance testing service has been used extensively by members and an interoperable solution has been achieved in many occasions with support from the OpenECG Helpdesk. In the period 2003-2006, more than 1700 ECGs were submitted for conformance testing by more than 20 countries worldwide (see Figure 5). Leading is Italy with 11 members, who have submitted 37.3% of the tests. After Italy, most ECGs have been submitted by Greece (17.74%) and Hungary (11.89%). As SCP-ECG is a European standard, 92% of the ECGs were submitted by Europe, 4% by Asia (Japan, Taiwan, China, India, Hong Kong), and just 3% by USA. ECG devices and eHealth services have been tested, improved, and validated using online tools and support from the OpenECG Helpdesk. In addition, based on member feedback and analysis of tests carried out, the robustness of the service is constantly improving. Several integrators have received an OpenECG interoperability certificate.

3.3 OpenECG Open Source Repository

The lack of publicly available OSS for ECG acquisition, storage, viewing, and serial comparison has been identified as one of the main reasons for the poor diffusion or the inaccurate implementation of ECG interoperability standards. OpenECG adopted an OSS strategy to complement the interoperability certification service and further assist developers in the correct implementation of ECG standards.

Main manufacturers and research centres were contacted for contributions to the OpenECG open source repository.

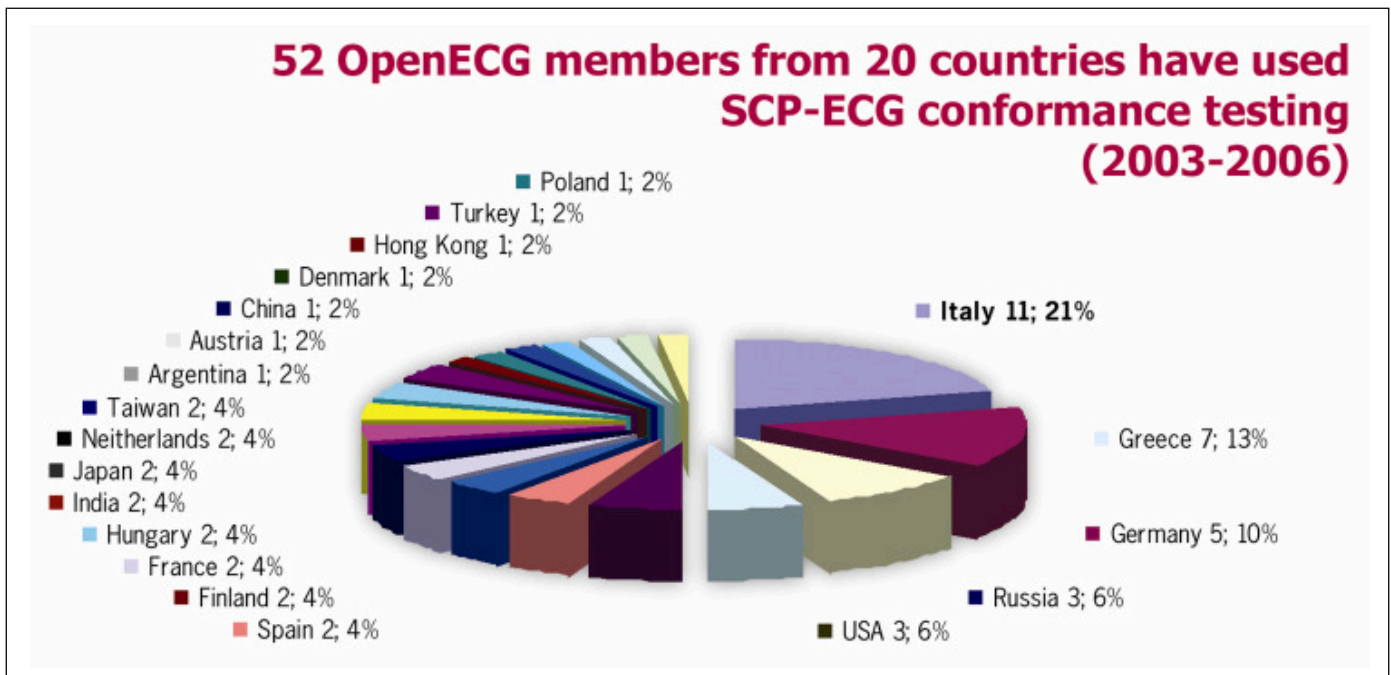


Fig 5 Use of the online conformance testing service for the SCP-ECG standard

Physionet [31] at MIT was very supportive and contributed a reader (i.e. parser) for a particular SCP-ECG dialect. Other major contributions to the repository were the submissions to the OpenECG programming contest and an SCP-ECG library in JAVA to read and write ECG records.

The OpenECG programming contest was organized to stimulate the creation of testing tools, converters, and SCP-ECG viewers. The contest evaluation criteria were SCP-ECG feature coverage, accuracy, ergonomics, readability and portability (cross-platform based only on OSS software tools). Winning contributions were: (a) SCP-ECG data reader/writer and C++ viewer with anti-aliasing signal representation [32]), (b) DICOM Toolkit extended with SCP-ECG viewing capabilities [33], and (c) SCP-ECG reader in Matlab/Octave [34]. The submissions to the OpenECG programming contest were tested with multiple SCP-ECG data sets. All these submissions to the OpenECG contest

are available in the OpenECG open source repository and serve as the basis for further development. For example in 2005, a web service variant of the conformance testing service was integrated to the ECG viewer in C++ and in that process certain limitations of the viewer were identified and amended [35].

The SCP-ECG library in JAVA provides methods for reading and writing an SCP-ECG record covering signal redundancy suppression (2nd derivative) and Huffman encoding. The library was tested with normative SCP-ECG samples from the OpenECG portal and ECG records from six different ECG device vendors.

OSS in the OpenECG repository can be freely downloaded, redistributed and used as the basis for further development, under the terms of GNU General Public License [3].

Already the SCP-ECG library has provided SCP-ECG support to several eHealth applications. Firstly, an **ECG management system** built in the IFC-CNR hospital receives ECG records

from different models of electrocardiographs and stores them in a specific file folder monitored by a software daemon. Each new ECG is processed by the daemon and stored in the cardiology information system maintaining the EHR of the patient. Secondly, an **eMail SCP-ECG parsing service** build on the same library provides visual and textual feedback on the form and contents of SCP-ECG records. OpenECG members that have registered to receive the service, may send email to *scp-parser@ifc.cnr.it* with attached SCP-ECG files. Each email with an SCP-ECG attachment is automatically processed. A textual report with the contents of the SCP-ECG record and an image of the raw data signals is automatically sent back to the sender. The ECG device printout can be compared with these reports to provide feedback on the interoperability of the recorded ECG. Finally, in cooperation with a small Italian ECG device manufacturer an **eHealth application for the real-time acquisition,**

storage, processing, and communication of ECGs in the SCP-ECG format has been developed. ECGs are displayed in real-time, and diagnostic codes can be added to the SCP-ECG record, while customizable textual and graphic ECG reports can be printed.

Other notable cases include an ECG manufacturer and two Italian regional health authorities in Italy (ASL6 Livorno and ASL4 Chiavarese) that expressed interest in implementing SCP-ECG-compliant eHealth services based on OSS components from the OpenECG repository.

An experimental study was designed in collaboration with ASL6 Livorno to establish clinical data exchange including signals and images among a selected group of general practitioners' (GPs) practices, cardiologists and outpatient clinics. Such services are not particularly popular among GPs and that is not only due to political, ergonomic or technical reasons, but also to the lack of tools for easy management of ECG signals and patient data. The objective was to deploy a friendly network-integrated ECG service, using SCP-ECG compliant electrocardiographs and ECG e-mail, as well as software for ECG acquisition, display, and communication [36].

Occasionally, ECG manufacturers download the OSS software from the OpenECG repository to validate the implementations of SCP-ECG in their products. Sometimes they encounter problems and request assistance from the management of the OpenECG repository, but, with the assistance of the developers, the manufacturers are able to successfully use the tools at the OpenECG repository

At some time, another ECG manufacturer requested assistance to support ASL4 Chiavarese in assuring the conformance of ECGs to the SCP-ECG standard, in the context of an ECG overreading workflow. An ECG is re-

ceived by a specific cardiograph model and is stored as BLOB in an ECG table of an Oracle database to be later retrieved and overread by a cardiologist. It was necessary to extend the SCP-ECG library with a Java method able to receive as input two SCP-ECG files and to produce as output the result of the identity check between the signal sections of the SCP-ECG files. In the next days the software authors were in contact directly with ASL4 Chiavarese and in less than one day the Java method was successfully implemented. Since then, ASL4 Chiavarese is using the software in production [37].

In general, the response of the community to the OSS software developed and maintained by the OpenECG repository is positive. However, deployment of the OpenECG OSS by users in most cases required support by experts in the OpenECG community or the authors of the software, indicating a basic trend and common observation in OSS development.

4 Open Source as a Driver for Interoperability

Today the need for interoperable healthcare applications and tools is more imperative than ever. The complex workflows for the provision of health care in a modern clinical environment requires the presence of many potentially heterogeneous software entities that work in tandem and are able to understand each other. Open source as a methodology and development process can be also used in order to leverage interoperability and integration of heterogeneous software components. Open source software development alone does not guarantee the interoperability of the final products [38]: It is undoubtedly feasible, and it has

been the case in the past, to have different OSS products that are not able to exchange and use information from each other. In principle, interoperability is a goal that can be achieved by following open standards and widely accepted practices and methodologies. Nevertheless OSS can **foster interoperability and the development of standards that promote it**. The existence of an **open source reference implementation** for an open standard and specialized tools lower the cost of implementing standards and facilitate interoperability. From the developer's or the integrator's point of view, a large collection of OSS components is vital. With a reference implementation in open source it is easier to **build a community** around an open standard and assist in the standardization process with feedback from implementation.

This case was clearly demonstrated in the case of the OpenECG repository. The role of OSS in the initial development, evolution and adoption of the DICOM standard is another illustrative example for this case. The development of the "central test node" (CTN [39]) software by the Electronic Radiology Laboratory of the Mallinckrodt Institute was the first implementation of the written technical specification of the DICOM standard. This implementation served for many years both as a reference implementation of the DICOM standard and as a conformance test suite which vendors could use in order to test the compliance of their implementations to the DICOM standard. Since 2000, the OFFIS DICOM DCMT offers a collection of libraries and applications implementing large parts of the DICOM standard for medical image communication and serves as a reference implementation in Europe [40]. DICOM open source developments are complemented with the IHE Medical Enterprise Simulators and Analysis

(MESA) tools which provide support for conformance testing and eventually interoperability [41]. Thus, OSS **reduces the entry costs for the adoption of standards** and provides insight in the development of standards-compliant software. Moreover, OSS supports the development and evolution of standards specifications by providing freely available implementations that can be studied, inspected, modified, and experimented with. This was the experience reported in PICNIC for the development of services for RHINs and OpenECG in integrating electrocardiographs to EHR and eHealth applications using the SCP-ECG standard.

On a related note the espousal of open standards and open source implementations of these standards can avoid what is commonly referred as “vendor lock-in”. The use of proprietary software that does not adhere to open formats and standards can lead to a situation where a customer is dependent on a specific vendor’s offerings and switching to another product entails significant costs. In the public sector, it is not perceived as sustainable to allow a single vendor to dominate the market and the wide use of open data formats is a requirement for interoperability and the key for hampering single vendor monopolies. An example of such an open standard is the Open Document Format (ODF) which is a format for the storage and exchange of “office” documents such as text documents, spreadsheets, and presentations. ODF has been ratified as an OASIS Standard and also an ISO and IEC International Standard (ISO/IEC 26300:2006) and there are many open source and closed source software packages that support it, like the OpenOffice.org [42], Google Docs [43], StarOffice, etc.

The advantages offered by the open source methodology have been widely

acknowledged. Currently there are several successful initiatives around the world that promote open source as a facilitator for the diffusion of standards. The OpenEHR foundation is an international non-profit group collaboratively working to bring about comprehensive electronic health records [44]. The openEHR foundation is committed to supporting legislative and industry standards and works closely with standardization bodies as well as national and international project teams. Likewise, the Open Source Health Care Alliance [45] is an international collaborative promoting the development and use of OSS in the health care domain. The International Medical Informatics Association (IMIA) and the American Medical Informatics Association (AMIA) have established open source working groups that aim to provide education about the benefits of open source and raise awareness on OSS in general. Under the auspices of these organizations and the active members of the relevant user and developer communities, a substantial number of OSS packages have been developed. These OSS packages cover many different software needs in the healthcare domain ranging from image visualization, storage, and retrieval systems such as dsm4che -a DICOM implementation in Java [46], DVTK -a DICOM validation toolkit [47], and OSIRIS -a multiplatform DICOM viewer [48], to EHR software such as GNUmed [49], FreeMed [50], MirrorMed [51], OpenEMed [52], OpenEHR [53], and Care2X [54].

4.1 OSS in Retrospect: Advantages and Disadvantages

Frequently, health care organizations adopt open source projects to lower the total cost of ownership for technology, to access source code, to gain flexibil-

ity, to take control of modifications, and to obtain better security capabilities.

Commercial software packages are developed by skilled engineers inside the company under strict deadlines and rigid specifications. Frequently interoperability and conformance to standards are low on the list of features to be supported. Often the main company is supported by partners, small companies or group of users, generally for importing specific knowledge and for experimental product evaluation. The relationships between the company and its partners are always fixed by a legal agreement and the product is generally protected by proprietary licence. Nobody can manipulate the software and the source files, when available, are protected by a non-disclosure agreement. Once the product specifications are defined, they are very difficult to change. Any bug report and any request for software modification or customization has to be addressed to the company; and the user has to wait for the new release. Doug Schmidt, who leads the long term project on ACE+TAO, a successful open source middleware broker [55], notes that open source should be a facilitator to a business model to leverage expertise and improve software quality. He also notes that while a core group of ten good programmers can develop high quality software, for software testing you need a large number of programmers. Moreover, the turnaround time for correcting an error in OSS can be a lot shorter, as the person identifying the bug may also provide the fix (Figure 6). This was actually reflected in the experience of the OpenECG repository discussed in section 3.3 where all requests on OpenECG software customizations received a rapid response.

In the case where software vendors go out of business, health care institutions are left with a system that they spent a

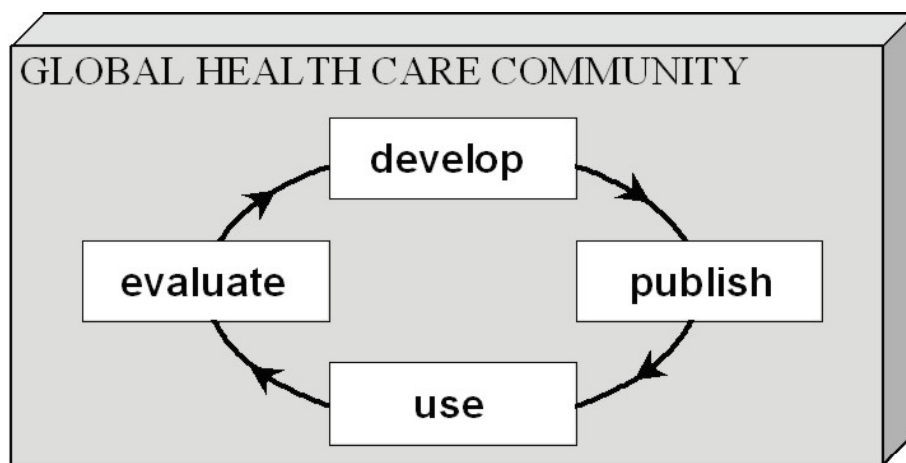


Fig. 6 OSS life cycle

lot of money to acquire and they cannot upgrade or maintain because there is no access to the source code. Security and reliability are enhanced when the source code can be inspected for bugs. OSS has the **potential for greater reliability** because anyone can study the source code, and “all bugs are shallow to a million eyes” [56], but this potentiality does not necessarily translate into reality. There is a huge controversy between proprietary and OSS, which is linked to open standards and the fallacy that medical software or hardware vendors may control one’s data and lock-out competitors at will.

At the same time, OSS has received a lot of criticism over time, as not all OSS exhibits the same level of quality and support. A close look at medical software available at SourceForge, the single largest repository of OSS on the Internet, clearly illustrates this point. There are a few highly popular medical software packages with high level of support and astounding number of downloads. However, the largest number of OSS projects is incomplete, full of errors, and with limited support. Serious technical shortcomings, complexity and lack of core features, like security, eventually lead to scorn and

abandonment. In reality, as Henning states [57] “*procedural failures are the root cause of OSS technical failures*”. Procedures to guarantee a successful OSS implementation include standardization of best practice, technical validation through reference implementations of realistic complexity, certification, and last but not least building a community to support its evolution.

Although, as far as interoperability in health care is concerned, technical validation and official certification are key to achieving interoperability in practice, our experience with PICNIC and OpenECG suggests that there is still a long way to go before it becomes a reality. An optimistic message can be found in the interoperability strategy of ETSI [58], where technical validation, conformance testing and certification are recognised as important parts of the standards developing process [59].

5 Conclusions

Health information Systems interoperability requires the consistent implementation of open standards. For open standards to be consistently imple-

mented they have to be well-documented and their full specification should be publicly available. OSS has several intrinsic characteristics that enhance software quality and may act in support of interoperability. However, an aspect of OSS that can be a critical point to its wide adoption is the ability to form an OSS community with adequate capacity and knowledge to provide development support, sustain high software quality, and support OSS evolution.

In retrospect, OSS in health care contributes to the standards adoption process supporting interoperability. Health information systems benefit from an open source community which, in cooperation with SDOs, supports the maintenance of reference implementations to establish and further develop eHealth standards. In the long run, however, it is the commercial marketplace that acts as the test bed and evolutionary filter.

References

1. Templeton B. 10 Big Myths about copyright explained, 1994. Available on-line and continuously updated: <http://www.templetons.com/brad/copymyths.html> (accessed April 17, 2007).
2. Berkeley Software Distribution (BSD) License, <http://www.opensource.org/licenses/bsd-license.php> (accessed April 17, 2007).
3. Free Software Foundation. GNU General Public License. In: DiBona C, Ockman S, Stone M, editors. Open sources: voices from the open source revolution. Sebastopol, CA: O’Reilly and Associates, 1999, www.gnu.org (accessed April 17, 2007) and <http://www.gnu.org/copyleft/gpl.html> (accessed April 17, 2007).
4. Open Standards Definition, International Telecommunications Union, <http://www.itu.int/ITU-T/othergroups/ipr-adhoc/openstandards.html> (accessed April 17, 2007).
5. Chronaki CE, Chiarugi F. Interoperability as a Quality Label for Portable & Wearable Health Monitoring Systems. In: Nugent CD, McCullagh PJ, McAdams ET, Lymberis A, editors. Personalized Health Management Systems: the Integration of Innovative Sensor, Textile Information and Communication Technologies. IOS Press, 117/2005.
6. Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A

- Compilation of IEEE Standard Computer Glossaries. New York, NY; 1990.
7. International Standardization Organization for Standardization Technical Committee 215 (ISO TC 215), <http://www.iso.org/iso/en/stdsdevelopment/tc/TC.html> (accessed April 17, 2007).
 8. CEN Technical Committee TC251, <http://www.cen251.org/> (accessed April 17, 2007).
 9. American National Standards Institute (ANSI), <http://www.ansi.org> (accessed April 17, 2007).
 10. Digital Imaging and Communications in Medicine (DICOM), <http://medical.nema.org/> (accessed April 17, 2007).
 11. Systematized Nomenclature of Medicine (SNOMED), <http://www.snomed.org> (accessed April 17, 2007).
 12. Object Management Group – Healthcare Domain Task Force, http://healthcare.omg.org/Roadmap/corbamed_roadmap.htm (accessed April 17, 2007).
 13. Health Level Seven (HL7), <http://www.hl7.org> (accessed April 17, 2007).
 14. Integrating the Healthcare Enterprise Initiative (IHE), <http://www.ihe.net/> (accessed April 17, 2007).
 15. Healthcare Information and Management Systems Society (HIMSS), <http://www.himss.org/ASP/index.asp> (accessed April 17, 2007).
 16. Karla D and Forslund D. Open Source Health Systems. In: Demetriades JE, Kolodner RM, Christopherson GA, editors. Person-Centered Health Records - Toward HealthPeople, Series: Health Informatics; 2005. XXIV, ISBN-10: 0-387-23282-6.
 17. Katakakis DG. The PICNIC Story. In: Regional Health Economics and ICT Services: The PICNIC Experience. IOS Press, Volume 115 Studies in Health Technology and Informatics. p. 4-36, hardcover, ISBN: 1-58603-538-X, August 2005. On-line version: http://www.ics.forth.gr/~katakaki/publications/picnic_story2005.pdf (accessed April 17, 2007).
 18. CEN TC251. Health informatics - Standard communication protocol - Computer-assisted electrocardiography. ICS: 35.240.80 IT applications in health care technology, Reference number EN 1064:2005, www.cenorm.org (accessed April 17, 2007).
 19. The OpenECG portal, <http://www.openecg.net> (accessed April 17, 2007).
 20. Katakakis DG, Bruun-Rasmussen M, Pakarinen V, Piggott D, Saranummi N. PICNIC Technology. Included in "Regional Health Economics and ICT Services: The PICNIC Experience", IOS Press, Volume 115 Studies in Health Technology and Informatics, pp. 61-91, hardcover, ISBN: 1-58603-538-X, August 2005.
 21. Katakakis D, Tsiknakis M. Electronic Health Record. In: Akay M, editor. Wiley Encyclopedia of Biomedical Engineering. Hoboken: John Wiley & Sons, Inc. dx.doi.org/10.1002/9780471740360.ebs1440; 2006.
 22. OMG Clinical Observations Access Service (COAS), version 1.0, http://www.omg.org/technology/documents/formal/clinical_observation_access_service.htm (accessed April 17, 2007).
 23. OMG Person Identification Service (PIDS), version 1.1, http://www.omg.org/technology/documents/formal/person_identification_service.htm (accessed April 17, 2007).
 24. Schmidt D. TAO Developer's Guide Verison 1.1a, Object Computing Inc.; 1.1a edition ;2000.
 25. Badilini F. Freeware ECG Viewer for the XML FDA Format, Proceedings of the 2nd OpenECG Workshop, Berlin 1-4 April 2004.
 26. Digital Imaging and Communication in Medicine (DICOM) Supplement 30 waveform interchange, DICOM Standards Committee, Working Group 1 –Cardiac and Vascular Information, 1 Nov 1999, http://medical.nema.org/Dicom/supps/sup30_lb.pdf (accessed April 17, 2007).
 27. Hirai M, Kawamoto K. MFER - a Japanese Approach for Medical Waveform Encoding Rules for Viewer Design, Proceedings of the 2nd OpenECG Workshop, Berlin 1-4 April 2004.
 28. Jettkant B, Dultgen P. Micro System Data Format (MSD) for Interoperability between Wearable Devices. In Proceedings of the 2nd OpenECG Workshop, Berlin 1-4 April 2004.
 29. European Data Format (EDF): <http://www.edfplus.info/> (last accessed April 17, 2007).
 30. PhysioToolkit, open source software for biomedical science and engineering, <http://physionet.ph.biu.ac.il/physiotools/index.shtml> (accessed April 17, 2007).
 31. PhysioNet, the research resource for complex physiological signals: <http://www.physionet.org/> (accessed April 17, 2007).
 32. Cerversato E, D' Odorico G. Standard Communications Protocol for Computer-Assisted Electrocardiography Advanced Viewer, Proceedings of the 2nd OpenECG Workshop, Berlin 1-3 April 2004.
 33. Clunie D. Extension of an Open Source DICOM Toolkit to Support SCP-ECG Waveforms, Proceedings of the 2nd OpenECG Workshop, Berlin 1-3 April 2004.
 34. Schloegl A. SCP-Converter for Octave and Matlab, Proceedings of the 2nd OpenECG Workshop, Berlin 1-3 April 2004.
 35. Chronaki CE, Chiarugi F, Sfakianakis S, Zywiwicz C. A Web Service for Conformance Testing of ECG Records to the SCP-ECG Standard. *Comput Cardiol* 2005;32:961-4.
 36. Conforti F, Micalizzi M, Conte R, Ciani G, Ghione S, Marabotti C, et al. Open Source Software to make the Telematic ECG Exchange popular among physicians. Proceedings of MEDICON 2004, Ischia, Naples, Italy, August 2004.
 37. Lightwood S, Greco F, Lizzio V, Brignole M, Gigli G. NET4ECG: An ECG Digital Network. Proceedings of Computers in Cardiology 2005;32:439-442. Lyon, France, September 2005.
 38. Gates B. Building Software That Is Interoperable By Design, <http://www.microsoft.com/mscorp/exccmail/2005/02-03interoperability.mspx> (accessed April 17, 2007).
 39. CTS, <http://erl.wustl.edu/research/dicom/ctn.html> (accessed April 17, 2007).
 40. DCMTK - DICOM Toolkit, <http://dicom.offis.de/dcmtdk.php.en> (accessed April 17, 2007).
 41. MESA tools, <http://ihedoc.wustl.edu/mesasoftware/index.htm> (accessed April 17, 2007).
 42. OpenOffice.org suite, <http://www.openoffice.org/> (accessed April 17, 2007).
 43. Google Docs, <http://docs.google.com/> (accessed April 17, 2007).
 44. Kalra D. Electronic health record standards. In: Haux, R, Kulikowski C, editors. IMIA Yearbook of Medical Informatics 2006. Stuttgart: Schattauer; 2006. p. 136-44. ISBN 139783794525256
 45. Open Source Health Care Alliance, <http://www.oshca.org/> (accessed April 17, 2007).
 46. dcm4che: Open Source Clinical Image and Object Management <http://www.dcm4che.org/> (accessed April 17, 2007).
 47. DICOM Validation Toolkit, <http://dvtc.org/> (accessed April 17, 2007).
 48. OSIRIS DICOM viewer, http://www.sim.hcuge.ch/osiris/01_Osiris_Presentation_EN.htm (accessed April 17, 2007).
 49. GNUMed, <http://www.gnumed.org/> (accessed April 17, 2007).
 50. FreeMed, <http://www.freemed.org/> (accessed April 17, 2007).
 51. MirrorMed, <http://www.mirrored.org/> (accessed April 17, 2007).
 52. OpenEMed, <http://openmed.sourceforge.net/> (accessed April 17, 2007).
 53. OpenEMR, <http://www.openemr.net/> (accessed April 17, 2007).
 54. Care2X, <http://www.care2x.org/> (accessed April 17, 2007).
 55. Schmidt D. Rationale for the ACE+TAO Commercial support Model, <http://www.cs.wustl.edu/~schmidt/TAO-support.html> (accessed April 17, 2007).
 56. Raymond ES. The cathedral and the bazaar. *First Monday*, March 1998, vol 3, no 3.
 57. Henning M. The Rise and Fall of CORBA, *ACM Queue* 4:5, June 2006. <http://acmqueue.com/modules.php?name=Content&pa=showpage&pid=396> (accessed April 17, 2007).
 58. European Telecommunications Standards Institute (ETSI): <http://www.etsi.org> (accessed April 17, 2007).
 59. van der Veer H. Achieving Technical Interoperability - the ETSI Approach, ETSI White paper, Oct 2006. http://www.etsi.org/etsi_radar/whitepaper/wp_3.htm (accessed April 17, 2007).

Correspondence to:

Catherine E. Chronaki
 Institute of Computer Science, FORTH
 PO 1385, Vassilikia Vouton
 Heraklion, Crete
 71110 Greece
 Tel: +302810391691
 Fax: +302810391428
 E-mail: chronaki@ics.forth.gr