Creation and Amalgamation for Health Information Exchange Using Cloud Computing System Based on CDA

A. Merry Ida, Manikandan K, Kannan R, Aditya S Warrier
Department of Computer Science and Engineering, Vel Tech High Tech Dr Rangarajan Dr. Sakunthala Engineering College, Avadi, Tamilnadu
***Corresponding author: E-Mail: idamerry@gmail.com**

ABSTRACT

The deployment of EHR has helped improve quality of care and patient safety, but it requires interoperability between HRE at different hospitals. The Clinical Document Architecture (CDA) developed is a core document standard that is critical to ensure such interoperability. In except for in a handful countries hospitals are reluctant to adopt interoperable HIS due to its deployment cost. A problem arises even when more hospitals start using the CDA document format because the data scattered in different documents are hard to manage. This paper describe our CDA document creation and amalgamation Open API service based on cloud computing, through which hospitals are enabled to conveniently generate CDA documents. This CDA document integration system assimilates multiple CDA documents per patient into a single CDA document which the physicians and patients can browse the clinical data presented to them in a chronological order. This CDA document generation and integration system is based on cloud computing and Open API. This ensures that the developers using different platforms can use our system to enhance interoperability.

KEY WORDS: Health information exchange, HL7, CDA, cloud computing, software as a service.

1. INTRODUCTION

HEALTH information is defined as information pertaining to the health of an individual or health care provided to an individual and it can support of efficient processes for health care delivery (Sung-Hyun Lee, 2016). Electronic Health Record (EHR) is longitudinal collection of electronic health information for and about persons, where In order to ensure successful an operation of EHR, a Health Information Exchange (HIE) system need to be implemented (Lee, 2014). However, most of the HIS in administration have distinctive attributes and are commonly incongruent (Colomo-Palacios, 2014). Hence, standardization is essential for effective health information exchange between hospitals. Particularly, clinical archive institutionalization lies at the center of ensuring interoperability.

Health Level Seven has established CDA as a major standard for clinical documents. CDA is a document markup standard that specifies the structure and semantics of 'clinical documents' for the purpose of exchange. The first version of CDA was developed in 2001 and Release 2 came out in 2005 (Gorp, 2012). Many projects adopting CDA have been successfully completed in many countries (Yi, 2012). Active works are being done on improving semantic interoperability based on open EHR and CEN13606.

The support of more HIS's for CDA is required for assuring confidence in HIE interoperability. The structure of CDA being complex, the production of correct CDA document is tough to achieve without deep knowledge about its standards. Each hospital requires a separate CDA generation system because HIS development platform for hospital vary greatly. Most hospitals also refuse to adopt new systems unless it is absolutely necessary for them. Thus, excluding a few countries such as New Zealand and Australia, the adoption rate of HER is very low.

A recording of the diagnosis is generated as a CDA document when a patient is diagnosed at a clinic. This CDA document is shared with other clinics if the patient allows it. In India, the concept of a family doctor does not exist, this makes it common for a patient to visit a number of clinics. The exchange of CDA document is prompted: when a physician needs to study a patient's medical history; when referral and reply letters are drafted for a patient cared by multiple clinics; when a patient is in emergency and the medical history needs to be reviewed.

As the number of CDA documents increases the time taken by the medical professional also increases because he has to collect a huge amount of individual files. Thus the decisions to be made by the medical professional is delayed. The integration all of the CDA document into a single document aids the medical personnel in review the patient's clinical history conveniently in sequential request per clinical area and the subsequent care administration can be conveyed more. For now, the method to integrate all the various individual CDA documents doesn't exist yet and expecting a hospital to develop a CDA integration system is not practical.

This paper presents (a) a CDA document generation system that helps generate CDA documents in various platforms (b) a CDA document integration system to integrate multiple CDA documents scattered in different hospitals for each patient. This system has a number of advantages. Firstly, the system enables developers who specialize in Java, .NET, or C/C++ to continue working on their developer platforms is accessible through an Open API. The existing system in hospitals can simply extended rather than completely being replaced with a new system. Secondly, it is

unnecessary for hospitals to train their personnel to operate on CDA documents because that would be rendered redundant by the system. Documents in the CDA format are produced by the CDA generation service in cloud produces approved by the National Institute of Standards and Technology (NIST). Thirdly, existing EHR are more likely to consider adoption of CDA in their practices if this service is provided for free or low prices to hospitals.

The organization of this paper is as follows. Section 2 contains detailed explanations on the format of CDA document, cloud computing, and the overall architecture of the system being proposed. In Section 3, the efficacy of the proposed system and its contrasts to different HIE systems in various countries is provided.

2. MATERIALS AND METHODS

In this section the necessary techniques are provided in detail for the design, and the implementation of our CDA generation and integration system based on cloud computing is explained briefly.

The CDA document: The American Nation Standards Institute approved HL7 Clinical Document Architecture Release 2 (CDA R2) in May 2005. CDA is an XML-based document markup standard which specifies the structure and semantics of clinical documents, and facilitating clinical document exchanges between heterogeneous software systems is its primary purpose. A CDA document is chiefly divided into header and body. The header includes information about the patient, hospital, physician, etc. provided with a clearly defined structure. The body which contains various clinical data is more flexible than the header. Every piece of clinical data is given a code and allocated a section as defined in the Logical Observation Identifiers Names and Codes (LOINC). Depending on the purpose of the document different subcategories are inserted in a CDA document, and we the Continuity of Care Document (CCD) is chosen since it contains the health summary data for the patient and it enables interoperability. The notable data included in CCD are listed in Table.1.

Table.1. Data items in CCD Header and Sections in the CCD Body

CDA location	Data items
CDA Header	Document Information (creation time, template ID, language code, purpose) Patient's information (ID, name, gender, birth date) Author's information (ID, name, represented organization) Organization's information (name, address, phone number)
CDA Body	Payers Advance Directives Support Functional Status Problems Family History Social History Allergies Medications Medical Equipment Vital Signs Results Procedures Encounters Plan of Care

For the integration of the CDA document, the Korean Standard for CDA Referral and Reply Letters (Preliminary Version) format is chosen as the number of clinical documents generated is large. It has the identical structure as the CCD and the types of data contained in the body are listed in Table.2.

Cloud computing: Cloud computing is a term which refers to the hardware and the applications delivered as services over the Internet and systems software in the data centers that provide those services. The user has to pay a fee depending on the amount of resources allocated to him, such as network, server, storage, applications and services. Currently there are three major types of cloud computing service which are:

Table.2. Sections in the Korean Standard for CDA Referral and Reply Letters Body (Preliminary Version)

Sections in CDA body	CDA Referral letter	CDA Reply letter
Diagnosis	No	Yes
History of past illness	No	Yes
History of Medication Use	Yes	Yes
Laboratory studies	Yes	Yes
Radiology studies	Yes	Yes
Pathology studies	Yes	Yes
Function Status Assessment	Yes	No
Surgical Operation Note	Yes	No
Relevant Diagnostic Tests	Yes	Yes
Reason for referral	Yes	No
Special Treatments and Procedures	Yes	No
Subsequent Evaluation Note	No	Yes
Plan of Treatment	Yes	Yes

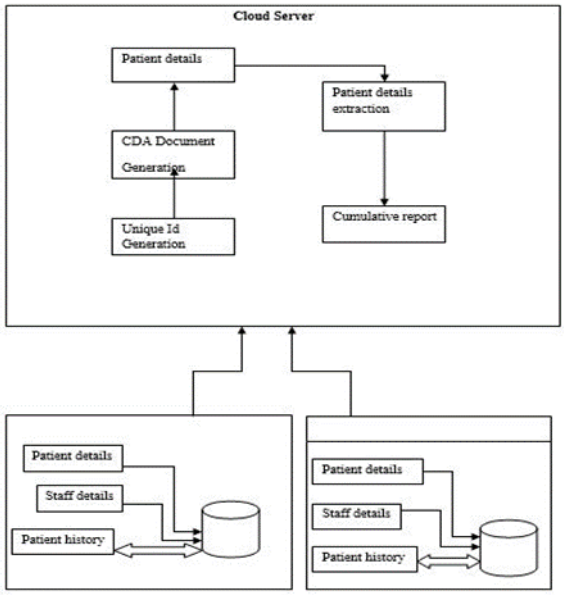

Figure.1. The architecture of our CDA generation system based on cloud computing

Table.3. Sections in the Korean Standard for CDA Referral and Reply Letters Body (Preliminary Version)

Sections in CDA body	CDA Referral letter	CDA Reply letter
Diagnosis	No	Yes
History of past illness	No	Yes
History of Medication Use	Yes	Yes
Laboratory studies	Yes	Yes
Radiology studies	Yes	Yes
Pathology studies	Yes	Yes
Function Status Assessment	Yes	No
Surgical Operation Note	Yes	No
Relevant Diagnostic Tests	Yes	Yes
Reason for referral	Yes	No
Special Treatments and Procedures	Yes	No
Subsequent Evaluation Note	No	Yes
Plan of Treatment	Yes	Yes

- Software as a Service (SasS): This service model provides software to the clients.
- Platform as a Service (PaaS): Cloud providers supply a computing platform where the clients can deploy applications of their own, program language of their own.

- Infrastructure as a Service (IaaS): Here the vendor integrates basic infrastructure like IT systems and database and then rents it out to a client.

In this paper, a widely used cloud service, Amazon Cloud, is chosen and the CDA generation and integration system as SaaS.

CDA Generation System Based on Cloud Computing: In Fig.1, how the architecture of CDA documents can be generated on the health information systems of different hospitals by using the cloud computing-based CDA generation system is shown.

The two hospitals A and B are demonstrated to show how easy it is to generate CDA documents on a variety of platforms if done using cloud. The purpose of each of the components is:

- CDA Generation API generates CDA documents on cloud server.
- CDA Generation Interface receives CDA documents generated in the cloud and uses the API provided by the cloud and also relays the input data.
- The template Manager is responsible for managing the CDA documents that is generated in the cloud.

CDA Generator is responsible for collecting patient data from hospitals and generating the CDA documents in the template formats as the Template Manager suggests.

- CDA Validator inspects if the CDA document which is generated complies with the CDA schema standard.

The DBMS at each hospital and the HIS are linked as follows. Hospital A, which uses a .Net-based system is connected via ODBC to connect to the DBMS while Hospital B, which uses a JAVA-based system, is linked with Hibernate.

In a hospital, the patient clinical information, hospital, and physician is entered via CDA Generation Interface and sent to the cloud server via CDA Generation API. We utilize SOAP (Simple Object Access Protocol) as transmission protocol for the purpose of enhancing interoperability among different HIS when a hospital sends data to the cloud. CDA Generation API is used to relay the data in the CDA Header/Body. The items that are included in usual CDA Header are: Patient ID, Date of Birth, Gender, First Name, and Family Name. In CDA Body, the following items are included:

Problem, Medication prescribed, Laboratory, Immunization provided, and so on. The data to be sent to the CDA Generation API are packaged as a whole in CDA Header Set and CDA Body Set and then relayed to CDA Generator. A CCD template from Template Manager is retrieved by the CDA Generator and the appropriate fields of the CCD template is filled in with the data from the CDA Header/Body Sets. The generated CDA document is inspected by the CDA Validator whether the CDA standards are being satisfied. It is inspected whether there is any missing element or the format is adequately followed. A CDA document is returned to the recipient hospital if no error is found. The different hospitals A and B have application that are coded in different development platforms to extend to generate CDA documents via a cloud server.

CDA Integration System Based on Cloud Computing: The standard for this integration of the various CDA documents is Korean Standard for CDA Referral and Reply Letters (Preliminary Version). The templates which generate a CDA use CCD part of Consolidated CDA which is released by ONC and made by HL7. However, an actually generated CDA has a form of CDA Referral and Reply Letters. The rationale for CDA document integration is as follows. When CDA-based (Health Information Exchange) is enthusiastically used among hospitals, the number of CDA documents concerning to each patient increases in time. For making a decision, the physician needs to spend a significant amount of their time to read these documents. In India the insurance model followed is fee for-service and therefore physician's consultation time spent per patient is very short since. Chronic patients especially are very likely to have been consulted by multiple physicians, in different hospitals. In this case, CDA documents may be scattered in different locations. Therefore, multiple CDA documents needs to be integrated into single CDA document. If the medical history of a patient is available in a single CDA document, the physician's time can be more efficiently used. This is evident when a patient is being referred to a different hospital or when a referral reply letter is sent. Our survey of physicians shows that displaying each section in chronological order helps improve the quality of care. This paper shows how we integrate CDA documents on a cloud server so that a variety of existing systems can be easily extended to generate integrated CDA documents.

At a hospital application, the CDA documents that are to be integrated are processed through the CDA Integration API of ours. The CDA Integration Interface then relays each CDA document sent to the cloud to the CDA Parser, which in turn converts each of the input CDA document into an XML object and then analyzes the CDA header and the analyzed data is grouped by each patient ID. The CDA Document Integrator then integrates the provided multiple CDA documents into a single CDA document. Following the LOINC values that set apart each section in the CDA document the data in the same section in the document body are merged. The CDA Validator is used to integrate. CDA document is inspected for error in, and the result is returned as string to the hospital that requested CDA document integration. This is because the CDA Integration System and the CDA Generation System are separate entities, and a new CDA document is made

after document integration, hence it is necessary to determine whether the new document complies with the CDA document integration, especially whether there is any missing element, or the format is wrong. Error messages, if found, are returned. Then the received string is converted to a CDA document file and saved. CDA Validator validates the data based on the CDA schema. An error is generated when a required field has been left blank or the wrong data type has been used. Example: The CDA document generation time, 'effective Time,' needs to be set, at least, in the YYYYMMDD format such as 20140806.

3. RESULTS

In this section, we report the results concerning the implementation of CDA generation and integration system based on cloud computing

Construction of a Cloud Computing:

Environment and Deployment of CDA:

Generation and integration System Based on it: We chose Amazon Elastic Compute Cloud (EC2) as the cloud platform for our CDA generation and integration system. Microsoft Windows Server 2008 Base was selected as its operating system. We chose Singapore as the server location.

Java (JDK 1.6) was used for CDA document generation and integration system and Tomcat 6.0.26 was selected as the web server platform for service deployment. As discussed in Section 2, we developed the CDA document integration and integration system and deployed the system on the Amazon Cloud Server.

Hospitals conveniently generate and integrate CDA documents by exploiting the API offered by our system.

Generation of CDA Documents on Different Developer Platforms through Cloud: To verify whether the system functions as designed, we requested CDA document generation on multiple systems implemented on different developer platforms via our API. For input data, we used the sample patient data offered by the US EHR Certification Program, Meaningful Use. The data does not pertain to any actual person. It is fictional, and available for public access. The use case scenario and data for CDA document generation are shown in Table.3.

Fig. 3 shows the JAVA-based HIS (Health Information System) indicated in Fig.1. Fig.3 is a screenshot of our API when requesting a CDA document generation for a hypothetical hospital that uses Java as its developer platform. Fig. 4 is a screenshot of using the API of our service to generate a CDA document by a hypothetical hospital that uses C# as its platform. When the user clicks on the button 'Generate CDA,' the data in each tab is first transmitted to the CDA Generation API in the cloud server via CDA Generation Interface and a CDA document is generated.

Table.4. The Use Case Scenario and Data for CDA Document Generation

Patient name	Manikandan Kannan
Date of birth	28/8/1995
Provider's name	Dr. Kannan Ramadass
Provider's office contact information	555–555–1002, Get Well Clinic, 1002 Healthcare Dr. Portland, OR 97005
Reason for Visit/Chief Complaint	High Fever, 3 days Chills, 4 days Cough with production of yellow- green sputum, 1day
Smoking Status	Current every day smoker, Start: 2014
Medications Administered During Visit	Albuterol 0.09 MG, 3 puffs once
Problems	Pneumonia; Asthma; AIDS; Hypoxemia
Laboratory Tests and Values/ Results	HGB 14.2; HCT 55%; WBC 7.8 (10^3/ul); PLT,223 (10^3/ul)
Immunizations	Tetanus - diphtheria adult, 1/4/2015 Influenza virus vaccine, 11/1/2013

We use the CDA document validation tool provided by US NIST to verify the validity of our CDA documents. (http://cda-validation.nist.gov/cda-validation/validation.html), which has the authority to certify CDA documents, to validate the CDA documents generated by using the API at our cloud server. The CDA documents generated by two clients developed with Java and C#, respectively, passed the validity test.

- Integration of CDA Documents via Our Cloud Server

We integrated multiple CDA documents of patient referrals and replies by using the API at our server. The use case scenario and patient data used for integration are shown in

Table.5. The Use Case Scenario and Patient Data Used for Integration

Patient Characteristics	The patient is a 20-year-old Indian male with a history of asthma which was controlled by albuterol for breakthrough. 8/6/2012
Diagnosis	pneumonia
History of Medication Use	albuterol, inhaled, Inhalant/Respiratory, 2.5mg/3ml NEB 3 times daily PRN wheezing/ shortness of breath ceftriaxone, IV, 1 gram IV once daily, 08/15/2012
Diagnosis	Costal Chondritis
Procedures	Pulmonary function tests
Laboratory Studies	CO2, 23 mmol/L 08/15/2012
Diagnosis	Angina
History of Medication Use	Aspirin 85 MG Oral Tablet, twice daily
Procedures	EC Procedure Intranasal oxygen therapy
Functional Status Assessment	Memory impairment Dependence on walking stick
Laboratory Studies	Na, 150 mmol/L

K, 5.0 mmol/L; Cl, 102mmol/L; CO2, 27mmol/L; BUN, 23 mg/dL; Cr 1.4 mg/dL; Glu, 202 mg/dL, Troponin T, 0.03 ng/ml.

The data given above does not pertain to an actual person. It is fictional, and available for public access.

4. DISCUSSION AND CONCLUSION

Not only is patient safety and quality of care improved by interoperability between hospitals but also time and resources spent on data format conversion is reduced. Interoperability is treated more important as the number of hospitals participating in HIE increases. If even one hospital does not support interoperability, the other hospitals have to convert the data format of their patient information to exchange data for HIE. Complexity for HIE unavoidably increases in proportion when the number of hospitals that do not support interoperability.

Changing an existing system adds cost for software and maintenance and hence hospitals are reluctant to adopt EHR systems that support interoperability. The advantages of an API service as ours are at the amount of resources that hospitals need to allocate for interoperability is minimal. Therefore, offering a system is a good alternative for hospitals that supports interoperability with cloud computing that have not yet adopted EHR because of cost issues.

The CDA document format is a clinical information standard designed to guarantee interoperability between hospitals, a large number of HIE projects that use the CDA document format have been undertaken in many countries.
Our cloud computing based CDA generation and integration system has a few distinct advantages over existing projects. First, hospitals do not have to purchase propriety software to generate and integrate CDA documents they produce. Second, our service is usable by various developer platforms because an Open API is to drive our CDA document generation and integration system. CDA documents can be easily generated to support interoperability in spite of the type of the platform,. Thirdly, CDA document generation and integration system which is based on cloud server is more markedly useful over existing services for CDA document if the variety of CDA document present increases. As of December 2013, there are 54 different types of CDA documents recognized by US NIST, and the number continues to grow year by year. With the cloud-based architecture proposed in this paper, it becomes convenient to generate documents that comply with new document standards. Thus, the cloud server can readily provide documents that comply with CDA Release 3 if only the server adopts its model, data type, and implementation guidelines.

As the number of HIE based on CDA documents increases, interoperability is achieved, but it also brings a problem where managing various CDA documents per patient becomes inconvenient as the clinical information for each patient is scattered in different documents. The CDA document integration service from our cloud server adequately addresses this issue by integrating multiple CDA documents that have been generated for individual patients.
The clinical data for the patient in question is provided to his/her doctor in chronological order per section so that it helps physicians to practice evidence-based medicine.

In addition, patients are enabled to use the CDA document integration service to obtain Personal Health Record (PHR) which contains not only clinical documents but also Personal Health Monitoring Record (PHMR) and Patient Generated Document (PGD). Patients can effectively generate and manage their PHR by using our cloud-based CDA document integration service.

The following problems were encountered while developing our CDA document generation and integration system. First, the default language of the Amazon Cloud OS is US English and it will not adequately handle Indian

language in the CDA documents. When SaaS is offered targeting hospitals of different languages, developers will need to pay extra attention to this issue. Secondly, our CDA document generation services API parameter for was of the list type, but under the C# language environment, the parameter was converted to the string array type. This is doubted to be caused by the IDE software of C#, which automatically makes this type conversion. Hence, the returned data needs to be as generic as possible to be made applicable to as many platforms as possible.

Future Advancements: The CDS integration system explained above has a number of drawbacks which are the following: 1) The system is rendered useless when the identity of the patient is unknown and cannot be quickly deduced in case of a life threatening situation 2) The data transfer and storage is not ensured to be secure and this poses a security threat to the patient's data on the cloud server. This confidential data in the hand of third parties with detrimental intentions can endanger the life of the patient.

Thus a finger print recognition or a password enabled validation system is envisioned to help overcome this obstacle that threatens the patient. A finger print enabled system would render any attack through the internet useless as the patient has to be present in person and in case of an emergency the details can be obtained just by using the patient's fingerprints.

5. ACKNOWLEDGEMENT

This paper is inspired from "Creation and Amalgamation for Health Information Exchange using Cloud Computing System based on Clinical Document Architecture." We would like to express our deepest gratitude and appreciation to our guide assistant Prof Mrs. Merry Ida A. We are thankful to her for the excellent guidance and encouragement she has provided us during the course of this project.

REFERENCES

Colomo-Palacios R, Stantchev V and Rodrıguez-Gonzalez A, Special issue on exploiting semantic technologies with particularization on linked data over grid and cloud architectures," Future Generation Comput. Syst, 32, 2014, 260–262.

D'Amore J.D, Sittig D.F, Wright A, Iyengar M.S and Ness R.B, The promise of the CCD: Challenges and opportunity for quality improvement and population health," in Proc. AMIA Annu. Symp. Proc, 2011, 285–294.

Gorp P.V and Comuzzi M, MyPHRMachines, Lifelong personal health records in the cloud, in Proc. 25th Int. Symp. Comput.-Based Med. Syst., 2012, 1–6.

Gorp P.V, Comuzzi M, Fialho A and Kaymak U, Addressing health information privacy with a novel cloud-based PHR system architecture, in Proc. IEEE Int. Conf. Syst, Man, Cybern, 2012, 1841– 1846.

Huang K, Hsieh S, Chang Y, Lai F, Hsieh S and Lee H, Application of portable cda for secure clinical-document exchange, J. Med. Syst., 34 (4), 2010, 531–539.

Kikuchi S, Sachdeva S and Bhalla S, Applying cloud computing model in PHR architecture, in Proc. Joint Int. Conf. Human Centered Comput. Environments, 2012, 236–237.

Kuperman G.J, Blair J.S, Franck R.A, Devaraj S and Low A.F, Developing data content specifications for the nationwide health information network trial implementations, J. Am. Med. Inform. Assoc., 17 (1), 2010, 6–12.

Lee S, Song J and Kim I, Clinical document architecture integration system to support patient referral and reply letters, Health Informat. J, Published online before print, 2014

Martınez-Costa C, Menarguez-Tortosa M and Tomas Fernandez-Breis J, An approach for the semantic interoperability of ISO EN 13606 and OpenEHR archetypes, J. Biomed. Inform, 43 (5), 2010, 736–746.

Narayanan H.A.J and Giine M.H, Ensuring access control in cloud provisioned healthcare systems, in Proc. IEEE Consumer Commun. Netw. Conf, 2011, 247–251.

Santos MR, Bax MP and Kalra D, Building a logical EHR architecture based on ISO 13606 standard and semantic web technologies, Studies Health Technol. Informat,160, 2010, 161– 165.

Sung-Hyun Lee, Joon Hyun Song, and Il Kon Kim, CDA Generation and Integration for Health Information Exchange Based on Cloud Computing System, 9 (2), 2016.

Yi S, Andrzejak A and Kondo D, Monetary cost-aware check pointing and migration on amazon cloud spot instances, IEEE Trans. Services Comput, 5 (4), 2012, 512–524.