A Relational Method for Determining Eventual Causality in Electronic Health Records

Bilal El-Hajj-Diab
School of Computer Science and Software Engineering
University of Wollongong
Wollongong, Australia
email: b.diab@alumni.uottawa.ca

Ali Hage-Diab
Department of Biomedical Engineering
Lebanese International University
Beirut, Lebanon
email: ali.hagediab@liu.edu.lb

Bassam Hussein
Department of Industrial Engineering
Lebanese International University
Beirut, Lebanon
email: bassam.hussein@liu.edu.lb

Mohamad Raad
Department of Computer and Communication Engineering
Lebanese International University
Beirut, Lebanon
email: mohamad.raad@liu.edu.lb

Hassan M. Khachfe
Center for Quality Assurance, Institutional Assessment &
Scientific Research (QAIASR)
Lebanese International University
Beirut, Lebanon
email: hassan.khachfe@liu.edu.lb

***Abstract*— The use of electronic health records (EHR) has been increased substantially to improve quality of healthcare outcomes. That's why recent EHR systems have been eagerly evolving from patient documentation systems toward intellectual tools for physicians to accomplish their tasks. Such evolution would mean continuous changes to the current EHR systems, which would be a real challenge since the designed software for these systems is usually customized based on prior requirements and thus having the correct workflow design is an essential key to allow system intellectuality and system automation. Such upgrade would give more realistic view to the patient medical profile by associating all related medical entities in a hierarchical order without losing the usability and performance of these systems. This new approach treats the EHR system as a storage manager, which is an important advantage from the development prospective especially for any future changes in any of the different layers.**

Keywords-workflow; electronic health record; causality; logical modeling.

I. Introduction

Consider a simple chat regarding a missing boy; the mother who is worried about her child not coming home sends a message to her friend asking "my boy is missing" (message 1), then the mom sends another message stating that her boy is home and no need to worry (message 2); her friend replies saying "thanks god" (message 3). So for these three messages it's obvious that "message 3" was caused by " message 2" and the correct workflow of causality of these messages concluded that the "thanks god" message makes sense since now the boy is safe and she is happy about it. Now, assume that for some reason "message 2" is missing then the whole concept would change and therefore "message 3", which is "thanks god" would look as if it was caused by the "message 1" and in this case the message would be completely misinterpreted because "thanks god" would be an answer to "my boy is missing" message.

In medical profiling, similar scenarios may apply but in much more serious conditions [1]. Consider a patient who had a minor head injury two days before going to a clinic for a severe swelling sinus and headache. After several examinations by the physician, he was diagnosed with flu. The head injury was not considered since it had nothing to do with his symptoms so it was ignored. After few years, the same patient was taken to the clinic again with serious symptoms of amnesia. After careful examination, the doctor concluded that this patient might be developing Alzheimer disease. However, there could be a possibility that the minor head injury which occurred a couple of years ago had something to do with this patient memory loss, but since this fact was not eliminated before, neither the doctor nor the patient would even consider such possibility. The flow of the symptoms and diagnosis can be thought of as messages or processes of causality. For example, head injury is "message 1", then flu symptoms is "message 2", followed by amnesia, which is "message 3". Therefore, if we had a knowledgebase to remove message 2 from the flow, then it would have been clear to the physician that amnesia (message 3) might have been caused by the head injury (message 1).

In this paper, we introduce a causal workflow protocol, which will act as a shim. A shim is a library that transparently intercepts application program interface (API)

 ISBN: 978-1-61208-511-1

calls on top of the electronic health records (EHR) system and produces symptoms, as well as diagnoses based on causality to assist physicians to have as much as possible accurate presentation of the patient's electronic profile.

Therefore, a bolt-on causal workflow layer to be added on top of a general purpose EHR system will be responsible to enforce the constraints for association and relation between the patient medical entities and would leave all other functionalities of the EHR management intact. The bolt-on will act as a special task middle layer between the user and the system by dealing with EHR system as a storage system in most of the cases. Medical diagnosis information processing dates back to the early seventies [2][3].

Section II outlines the method and implementation procedure. Section III presents enhancement recommendations for future work. Section III concludes the article.

II. METHOD AND IMPLEMENTATION

In general, any typical medical procedure would consist of a set of sequential examinations in order to determine the diagnoses [4]. We can think of these procedures in EHR system as top to bottom hierarchical workflow tree (t) while every process or examination would be considered a node (n) in this tree of flow since every visit (v) would be considered a set of examination as illustrated in Fig. 1. Therefore, every patient visit would create a new tree of examination workflow consisting of a set of processes that may be viewed as nodes.

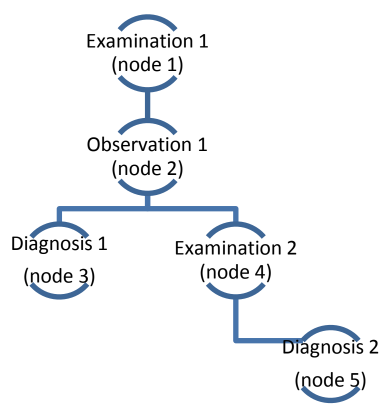

Figure 1. Patient visit process workflow.

Where the node n denotes a medical or process that can affect the decision of the practitioner and therefore would affect the process tree. Now, for m visits, a patient would have:

$\mathbf{tv_1 \ldots tv_m}$
of workflow trees
where each tree consists of
$\mathbf{n_1 \rightarrow n_2 \rightarrow n_3 \rightarrow \ldots}$

Noting the sequential relation between the nodes in every workflow, almost every patient visit to a doctor can be related to a previous condition from previous visit. Hence, there would be a cross relation between the workflow trees such that $tv_1 \rightarrow tv_2 \rightarrow \ldots$ or in another word causal relationship.

Clearly, we can see that with a missing causal reading consistency in the EHR system, the patient workflow trees would not provide the exact history and causality and therefore enormous work would be up to the physician to handle manually, redundantly, or even worse to build an assumption based on missing relations. This becomes an important operational problem [5].

The bolt-on would have customized consistency rules considered as meta-data. It would always put a unified causal tree of related processes, which the client can only see without further development. This shim would check constraint on write and read based on causal-relation of any process or examination, which in turn would guarantee having all sequential nodes returned even across different trees (horizontally in same workflow and vertically across other workflows) and provide them seamlessly and cohesively as depicted in Fig. 2. Finally, the architecture suggests layers separation of safety and live-ness concerns. So while the shim layer would handle consistency and visibility, the underlying EHR would handle the live-ness and usability to the end user.

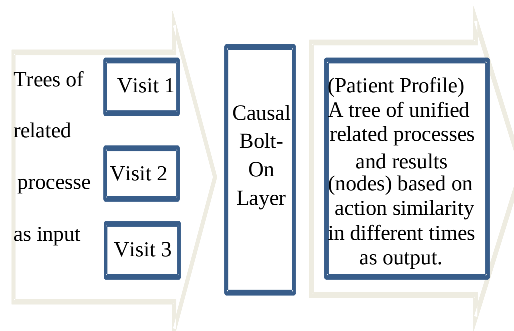

Figure 2. Bolt-on design unified causal tree.

After each patient visit, physicians would only be concerned about the diagnosis of the patient in that specific event and therefore all symptoms and incidents leading to the conclusion to a certain diagnosis would be ignored. As a result, symptoms and incidents would be observed only at the analysis level currently; however, the set of concluded diagnosis might be observed through any particular visit, although sequential observation is most common.

As shown in the above example, it's clear that valuable incidents or symptoms (process) could be overlooked in the time line of patient profile. Keep ALIVE methodology can be applied to incidents, which are candidates to be needed later. The need for each incident can be decided based on a scale from 1 to 10 concluded from a knowledge base providing the most-likability reaction between an incident, a symptom and a diagnose.

In order to better evaluate the process, we can convert the given example to a logical model. Logical models are actionable plans, strategies or maps with clear outcomes and

 ISBN: 978-1-61208-511-1

explicit steps for solving program problems. We use the logical model in planning, implementation, evaluation and communication. A more advanced notion of such models, such as fuzzy logical models has been proposed. They have contributed to building medical expert systems to assist physicians in medical decision-making and patient care [6].

In any particular patient examination or visit, the current symptoms and incidents will be treated as inputs, observations checkups and examinations as activities where diagnosis and final conclusions are the outputs. In addition the previous diagnosis which is the history of the patient would be considered as inputs also since it is pre-given and has a direct effect on the whole logical process.

The given example can be logically proposed in the logical model in Table I.

TABLE I. PATIENT FIRST VISIT LOGICAL MODELING.

Input	Activity	Output
- Current Symptoms o Headache o Sinus - Current Incidents o Head Injury - Previous Diagnosis	- Observation - Check-up - Examination	- Diagnosis o Flu - Prescriptions o Med1 o Med 2

In the second patient visit, the logical model would most likely look as shown in Table II.

TABLE II. PATIENT SECOND VISIT LOGICAL MODELING.

Input	Activity	Output
- Current Symptoms o Headache o Amnesia - Current Incidents o <u>None</u> - Previous Diagnosis	- Observation - Check-up - Examination	- Diagnosis o Alzheimer - Prescriptions o Med 1 o Med 2 o Med 3

Consider a given knowledgebase k1: Head Injury -> Brain bleeding -> Amnesia -> …

When a patient comes back with a symptom that belongs to a knowledge base and also a child of previously inspected incident, then that incident would be flagged as ALIVE in the database and it would then be added automatically as input to the current visit logical model. Therefore, the auto-adjust logical model would be shown as illustrated in Table III.

TABLE III. PATIENT DIAGNOSIS LOGICAL MODELING WITH CAUSALITY ASSOCIATION.

Input	Activity	Output
- Current Symptoms o Headache o <u>Amnesia</u> - Current Incidents o None - **<u>ALIVE Incidents</u>** o **Minor Head Injury** - Previous Diagnosis	- Observation - Check-up - Examination	- Diagnosis o Alzheimer - Prescriptions o Med 1 o Med 2 o Med 3

The suggested association would always be present as input based on the given medical knowledge base, which in turn would provide enough resources to accurately provide the patient present condition auto-linked with all previous related conditions or incidents denoted in table II underlined and in bold in the input section.

From an implementation prospective, this process requires a set of local resources such as linked knowledge base. However with the tremendous medical data available online now a day, making certain online resources in a distributed environment available as inputs to EHR input or activities sections in the logical model would clearly enrich the system and would keep practitioners on the track with the latest medical trends. That being said, it would be beneficial to encapsulate the implemented code in the form of web services. Creating web services would easily be used by different EHR systems written indifferent programming languages and would act as a standalone layer [7]. Web services can be described as applications which are dynamically self-contained, it can easily published, consumed and located from distributed applications or models over a local network or web based or even in a local client server applications. Web services are built on top of open standards such as TCP/IP (Transmission Control Protocol/Internet Protocol), HTTP (Hypertext Transfer Protocol), also Java in-addition to HTML (HyperText Markup Language), and XML (Extensible Markup Language).

Typically, web services are XML-based information exchange systems, which use direct end-to-end application interaction by exchanging XML messages and can preserve both security and performance [8].

III. FUTURE WORK

Adding ontology to the syntactical data entery would not just ease practioner work, but also would give the possibiltiy to auto analyze medical profiles and allow sophiscticated relations between various entities to be associated [9]. Such critieria would even enhance the medical systems at global bases where signs of epidemics would be detected at earlier stages. An ontology is defined as a formal, explicit specification of a shared conceptualization [10]. Several ontological frameworks for describing space and spatial relations have been developed recently [11].

In order to develop such add-on, natural language programming (NLP) would need to be applied. NLP is a component of artificial intelligence which gives the ability of a computer program to understand human speech as it is spoken. However, this is a very challenging approach especially that human speech is not always precise. In fact, it is often ambiguous and the linguistic structure can depend on many complex variables, including slang, regional dialects and social context. Due to the challenges of NLP, we aim to create a custom local grammar that has “high” end-user with XML based easy customization at the development level, which would allow adjustments with minimal coding.

 ISBN: 978-1-61208-511-1

IV. CONCLUSION

Our causal bolt-on will treat EHR system as a storage manager and that would be similar to creating a 3-tears application which is an important advantage from the development prospective especially for any future changes in any of the different layers. As a result, such improvement would guarantee seamless and yet very effective addition to the electronic health records regardless of the used platform toward an optimized electronic patient profile.

REFERENCES

[1] B. El-Hajj-Diab, I. Zayour, and A. Hage-Diab, “Auto-Learning Electronic Medical Record System”, 20th LAAS International Science Conference, Beirut, Lebanon, March 27–29, 2014.

[2] P. Wortman, "Medical diagnosis: an information-processing approach." Computers and Biomedical Research, pp. 315-328, 1972.

[3] E. Shortliffe and B. Buchanan, "A model of inexact reasoning in medicine." Mathematical Biosciences, pp. 351-379, 1975.

[4] S. Kumar and K. Aldrich, “Overcoming barriers to electronic medical record (EMR) implementation in the US healthcare system: A comparative study”, Health Informatics Journal, pp. 306-318, December 2010.

[5] M. Öztürké, A. Tsoukiàs, and P. Vincke, "Preference modelling", In Multiple criteria decision analysis: State of the art surveys, pp. 27-59. Springer New York, 2005.

[6] R. Seising, "From vagueness in medical thought to the foundations of fuzzy reasoning in medical diagnosis", Artificial Intelligence in Medicine, pp. 237-256, 2006.

[7] I. Zayour and H. Hajjdiab, "How much integrated development environments (ides) improve productivity?", Journal of Software, pp. 2425-2431, 2013.

[8] Q. Yu, X. Liu, A. Bouguettaya, and B. Medjahed, "Deploying and managing Web services: issues, solutions, and directions", The VLDB Journal—The International Journal on Very Large Data Bases, pp. 537-572, 2008.

[9] J. Hasan and B. El-Hajj-Diab, "BioFlow: A web-based declarative workflow language for Life Sciences", 2008 IEEE Congress on services-Part I. IEEE, 2008.

[10] T.R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?", International Journal of Human-Computer Studies, pp. 907-928, 1995.

[11] C. Hudelot, A. Jamal, and I. Bloch, "Fuzzy spatial relation ontology for image interpretation", Fuzzy Sets and Systems, pp. 1929-1951, 2008.

 ISBN: 978-1-61208-511-1