Digital Personalized Health and Medicine
L.B. Pape-Haugaard et al. (Eds.)

doi:10.3233/SHTI200160

Heimdall, a Computer Program for Electronic Health Records Data Visualization

Niels MARTIGNENE[a,1], Thibaut BALCAEN[b], Guillaume BOUZILLE[c], Matthieu CALAFIORE[d], Jean-Baptiste BEUSCART[a], Antoine LAMER[a], Bertrand LEGRAND[d], Grégoire FICHEUR[a], and Emmanuel CHAZARD [a]
[a] *Univ. Lille, CHU Lille, CERIM EA2694, F-59000 Lille, France*
[b] *Univ. Lille, EA2694, F-59000 Lille, CH St Quentin, F-02100 St Quentin, France*
[c] *Univ Rennes, CHU Rennes, Inserm, LTSI – UMR 1099, F-35000 Rennes, France*
[d] *Univ. Lille, CERIM EA2694, General practitioner, F-59000 Lille, France*

Abstract. *Introduction*. Electronic health records (EHR) comprehend structured and unstructured data, that are usually time dependent, enabling the use of timelines. However, it is often difficult to display all data without inducing information overload. In both clinical usual care and medical research, users should be able to quickly find relevant information, with minimal cognitive overhead. Our goal was to devise simple visualization techniques for handling medical data in both contexts. *Methods*. An abstraction layer for structured EHR data was devised after an informal literature review and discussions between authors. The “Heimdall” prototype was developed. Two experts evaluated the tool by answering 5 questions on 24 clinical cases. *Results*. Temporal data was abstracted in three simple types: events, states and measures, with appropriate visual representations for each type. Heimdall can load and display complex heterogeneous structured temporal data in a straightforward way. The main view can display events, states and measures along a shared timeline. Users can summarize data using temporal, hierarchical compression and filters. Default and custom views can be used to work in problem-oriented ways. The evaluation found conclusive results. *Conclusion*. The “Heimdall” prototype provides a comprehensive and efficient graphical interface for EHR data visualization. It is open source, can be used with an R package, and is available at https://koromix.dev/files/R.

Keywords. Electronic health records, Visualization, Timeline, Feature extraction.

1. Introduction

Electronic health records data usually contain structured data, such as administrative data, diagnoses (e.g. ICD10 codes [1]), procedures, administered drugs, medical devices, laboratory results, and non-structured data, such as free-text medical records and medical imaging. All of them are characterized by their temporal aspects (e.g. the date of a measurement, the period of a diagnosis, etc.), and relate to one patient. Structured data often take the form of multivalued qualitative variables, encoded using terminologies, and supported by a hierarchical tree of sections and codes.

[1] Corresponding Author: Niels Martignene, Faculté de Médecine, CERIM, F-59035 Lille cedex, France; niels.martignene@protonmail.com.

Data visualization is interesting within 2 contexts. “Transactional” visualization relates to visualization of one patient’s data, in order to provide the patient with appropriate care. “Decisional” visualization relates the visualization of several patient’s data, for management or research purposes [2,3], notably in the frame of data reuse.

In both contexts, the main objective of data visualization is to provide the user with a comprehensive view, and to enhance the mental process of feature extraction [4], by making it easy to detect temporal associations. For instance, in case of INR elevation and red cells decrease, a physician infers the patient encountered a hemorrhage. However, data complexity may overload the user, leading to interpretation errors [5].

Many visual interfaces have been developed for transactional EHR data visualization [6], such as *LifeLines* [7], *MultiMedia Stream System (MMSS) TimeLine* [8], *KNAVE-II* [9] , *TimeLine* [10] *HARVEST* [11], or other prototypes [12–14]. However, to our knowledge, such representation methods are rarely implemented in commercial hospital information systems. Some visual interfaces have also been developed for decisional EHR data visualization, such as *OutFlow* [15], *Care Pathway Explorer* [16], *EventFlow* [17], or other prototypes [18–25]. Many of them are dedicated to specific tasks. The objective is to propose a visualization method that could either be used for transactional or decisional visualization of EHR data, and to evaluate it.

2. Method

An abstraction layer for EHR data was first defined, based on an informal literature review, and discussions between authors, taking profit from their experience in clinical care and data reuse. The prototype was intended to be embeddable in C++ programs, R [26], and web pages.

To evaluate the tool, two medical experts (not involved in the development) were asked to retrieved medical information over 24 cases of acute kidney injury (AKI, according to the KDIGO criteria [27]). The 24 cases were represented through 3 interfaces: the classical interface (“classical”: 1 page per type of information, with tables), the manually optimized interface (“manual”: the same, but manually filtered, to present only relevant data), and the Heimdall interface (“Heimdall”: developed in this project, without any prior data filtering). Expert *A* used “classical” to evaluate records #1-8, “manual” for records #9-16, and “Heimdall” for records #17-24. Expert *B* used the same interfaces respectively for records #17-24, then #9-16, then #1-8. Both experts were asked to answer five questions for each case (“d0” relates to the date of AKI, which was intentionally not explicit): (Q1) Was a nephrotoxic treatment discontinued, at d0 or d1? (Q2) Was there a urine dosage of sodium and potassium, at d0 or d1? (Q3) Was there a search for proteinuria, at d0 or d1? (Q4) Was there a urinary tract imaging at d0, d1 or d2? (Q5) Was a there an ICD10 code of AKI? Number of errors and time required to answer were compared using an analysis of variance.

3. Results

3.1. EHR data abstraction

We identified 3 types of information: (1) *events*, which occur at a single point in time, and have a label and a date (e.g. a medical procedure, a document), (2) *States*, which are characterized by a label, a start date and a stop date (e.g. a hospital stay, a diagnosis), and (3) *Measures*, which are characterized by a date, a label and a value (with or without normality range; e.g. a laboratory result, a clinical parameter).

For each of kind of information, the label could be part of a hierarchical tree. In such case, the hierarchy folding had to result in the merging of the symbols. Moreover, additional information could be provided, e.g. the content for a document, which is a sort of *event*, or the unit for a laboratory result, which is a sort of *measure*.

3.2. Prototype development

Heimdall is a read-only visualization tool, developed in C++, using OpenGL, and Dear ImGui. It is available for MS Windows and GNU/Linux. It can be used through its R package, which enables data loading and terminology handling, and can also be integrated inside web pages (through WASM and WebGL).

Pieces of data relating to the same data type are grouped into a component (e.g. 3 values of blood potassium → a unique curve of kalemia). Each component follows a time axis from the left to the right. Components are designed to be stacked vertically.

Figure 1 shows the representation of components according to their type, and their behavior in case of combination (e.g. in case of terminology folding or temporal compression). *Events* are represented as triangles, *states* as transparent rectangles, and *measures* as curves. When folded, measures are transformed into red events in case of abnormal values (outside normality range), or blue events elsewise. The interpolation method can be set as Last Observation Carrier Forward, linear, or cubic spline.

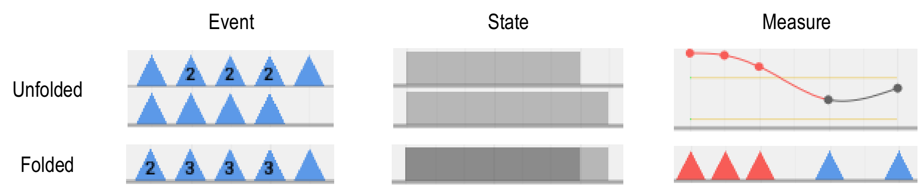

Figure 1. Unfolded and folded representation of 3 types of components

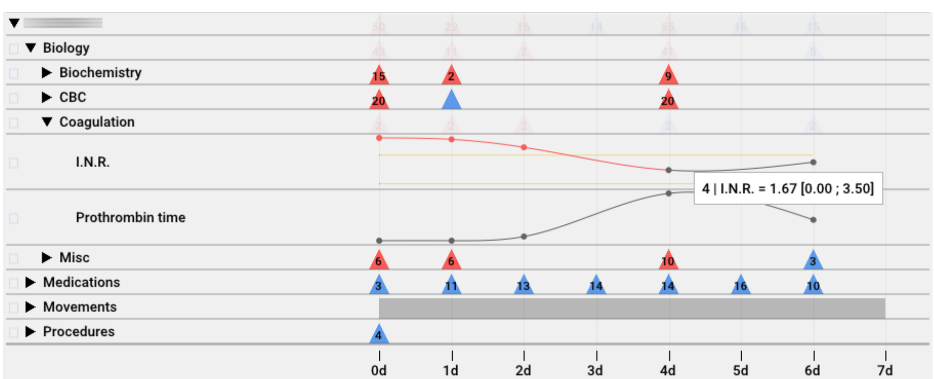

Figure 2. Example of a medical record (partially unfolded view)

Figure 2 illustrates a medical record. The main window principally looks like a stack of components, with a unique time axis. The components are organized according to their hierarchy: (1) patient, then (2) data type (e.g. diagnoses, laboratory results, etc.), then (3) the hierarchical tree of the relevant terminology, if so (e.g. ICD10). This organization is compatible with both transactional (one patient) and decisional (several patients) usage. In case of multi-patient view, the time axis can be aligned with real dates, or with a chosen event (e.g. birthdate, first inpatient stay, or specific surgical procedure). The representation of big amount of data is made easier by 3 mechanisms: (1) the time axis can be rescaled using control key plus mouse wheel, (2) the top-bottom axis can be compressed, resulting in components folding, and (3) a dropdown menu enables for quick use of problem-oriented filtered views. For instance, the “renal” view only shows diagnoses, procedures, laboratory results and drugs in relation with the kidneys or the renal function. The user can create custom views.

3.3. Prototype evaluation

Loading data from 3,500 patients, including 360,000 values, in Heimdall from R took a couple seconds on a workstation equipped with an old Intel® Core 2 Duo CPU (2009), and required about 50 MB of memory (40% for non-character data, 30% for character data, and 30% for terminologies/graphic interface/polices).

The results of the evaluation are presented in Table 1. For both users, the manually optimized interface and the Heimdall interface were faster to use than the classical interface. The number of errors did not significantly differ.

Table 1. Results of the evaluation (2 experts * 24 clinical cases * 5 questions)

Expert	Measure	“Classical”	“Manual”	“Heimdall”	*p val.*
#1	Time required (sec, mean & SD)	54.25 (5.42)	35.00 (2.62)	40.88 (5.22)	*< 0.001*
	Errors (n & %)	0.38 (0.74)	0.12 (0.35)	0.25 (0.46)	*0.662*
#2	Time required (sec, mean & SD)	57.75 (6.82)	42.25 (5.23)	45.38 (4.47)	*< 0.001*
	Errors (n & %)	0.38 (0.74)	0.12 (0.35)	0.12 (0.35)	*0.546*

4. Discussion and conclusion

In this work, we designed a data visualization prototype called “Heimdall”, and evaluated it. The Heimdall interface takes profit from combining foldable hierarchical view with a timeline. Alignment, filtering and custom views enhance data exploration. Heimdall is very fast and easy to use. However, this tool is still not able to display non-temporal data (e.g. birthdate), or data with imprecise timing (e.g. ancient medical history). Non-structured data can be displayed as document icons. Unlike EventFlow [17], Heimdall only provides rudimentary tools for querying and filtering data. The problem-oriented filtering seems to be the most powerful part. Heimdall provides a comprehensive and efficient graphical interface for EHR data visualization. It is open source, and the current R package (alpha version) can be downloaded at https://koromix.dev/files/R.

References

[1] World Health Organization, International Statistical Classification of Diseases and Related Health Problems, 2018. http://www.who.int/classifications/icd/en/ (accessed August 12, 2019).
[2] E. Chazard, and R. Beuscart, Graphical representation of the comprehensive patient flow through the Hospital, AMIA. Annu. Symp. Proc. 2007 (2007) 110–114.
[3] E. Baro, S. Degoul, R. Beuscart, and E. Chazard, Toward a Literature-Driven Definition of Big Data in Healthcare, BioMed Res. Int. 2015 (2015). doi:10.1155/2015/639021.
[4] E. Chazard, G. Ficheur, A. Caron, A. Lamer, J. Labreuche, M. Cuggia, M. Genin, G. Bouzille, and A. Duhamel, Secondary Use of Healthcare Structured Data: The Challenge of Domain-Knowledge Based Extraction of Features, Stud. Health Technol. Inform. 255 (2018) 15–19.

[5] H. Singh, C. Spitzmueller, N.J. Petersen, M.K. Sawhney, and D.F. Sittig, Information Overload and Missed Test Results in Electronic Health Record–Based Settings, JAMA Intern. Med. 173 (2013) 702–704. doi:10.1001/2013.jamainternmed.61.

[6] R. Bade, S. Schlechtweg, and S. Miksch, Connecting Time-oriented Data and Information to a Coherent Interactive Visualization, in: Proc. SIGCHI Conf. Hum. Factors Comput. Syst., ACM, New York, NY, USA, 2004: pp. 105–112. doi:10.1145/985692.985706.

[7] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman, LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records, in: B.B. Bederson, and B. Shneiderman (Eds.), Craft Inf. Vis., Morgan Kaufmann, San Francisco, 2003: pp. 308–312. doi:10.1016/B978-155860915-0/50038-X.

[8] D.A. Aoyama, J.-T.T. Hsiao, A.F. Cárdenas, and R.K. Pon, TimeLine and visualization of multiple-data sets and the visualization querying challenge, J. Vis. Lang. Comput. 18 (2007) 1–21. doi:10.1016/j.jvlc.2005.11.002.

[9] D. Goren-Bar, Y. Shahar, M. Galperin-Aizenberg, D. Boaz, and G. Tahan, KNAVE II: The Definition and Implementation of an Intelligent Tool for Visualization and Exploration of Time-oriented Clinical Data, in: Proc. Work. Conf. Adv. Vis. Interfaces, ACM, New York, NY, USA, 2004: pp. 171–174. doi:10.1145/989863.989889.

[10] A.A.T. Bui, D.R. Aberle, and H. Kangarloo, TimeLine: Visualizing Integrated Patient Records, IEEE Trans. Inf. Technol. Biomed. 11 (2007) 462–473. doi:10.1109/TITB.2006.884365.

[11] J.S. Hirsch, J.S. Tanenbaum, S. Lipsky Gorman, C. Liu, E. Schmitz, D. Hashorva, A. Ervits, D. Vawdrey, M. Sturm, and N. Elhadad, HARVEST, a longitudinal patient record summarizer, J. Am. Med. Inform. Assoc. JAMIA. 22 (2015) 263–274. doi:10.1136/amiajnl-2014-002945.

[12] W. Aigner, and S. Miksch, CareVis: Integrated visualization of computerized protocols and temporal patient data, Artif. Intell. Med. 37 (2006) 203–218. doi:10.1016/j.artmed.2006.04.002.

[13] FlowingMedia, TimeFlow: Timeline visualization application, 2018. https://github.com/FlowingMedia/TimeFlow (accessed April 18, 2018).

[14] N. Spring, C. Plaisant, G. Marchand, T.D. Wang, V. Mukherjee, D. Roseman, B. Shneiderman, and M. Smith, Temporal Summaries: Supporting Temporal Categorical Searching, Aggregation and Comparison, IEEE Trans. Vis. Comput. Graph. 15 (2009) 1049–1056. doi:10.1109/TVCG.2009.187.

[15] K. Wongsuphasawat, and D. Gotz, Exploring Flow, Factors, and Outcomes of Temporal Event Sequences with the Outflow Visualization, IEEE Trans. Vis. Comput. Graph. 18 (2012) 2659–2668. doi:10.1109/TVCG.2012.225.

[16] A. Perer, F. Wang, and J. Hu, Mining and exploring care pathways from electronic medical records with visual analytics, J. Biomed. Inform. 56 (2015) 369–378. doi:10.1016/j.jbi.2015.06.020.

[17] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman, Temporal Event Sequence Simplification, IEEE Trans. Vis. Comput. Graph. 19 (2013) 2227–2236. doi:10.1109/TVCG.2013.200.

[18] J.L. A, J.L. B, D.N. B, and E.K. A, VizTree: a Tool for Visually Mining and Monitoring Massive Time Series Databases, n.d.

[19] N.D. Soulakis, M.B. Carson, Y.J. Lee, D.H. Schneider, C.T. Skeehan, and D.M. Scholtens, Visualizing collaborative electronic health record usage for hospitalized patients with heart failure, J. Am. Med. Inform. Assoc. JAMIA. 22 (2015) 299–311. doi:10.1093/jamia/ocu017.

[20] J. Guerra-Gómez, M.L. Pack, C. Plaisant, and B. Shneiderman, Visualizing Change over Time Using Dynamic Hierarchies: TreeVersity2 and the StemView, IEEE Trans. Vis. Comput. Graph. 19 (2013) 2566–2575. doi:10.1109/TVCG.2013.231.

[21] R. Bellazzi, C. Larizza, P. Magni, and R. Bellazzi, Temporal data mining for the quality assessment of hemodialysis services, Artif. Intell. Med. 34 (2005) 25–39. doi:10.1016/j.artmed.2004.07.010.

[22] K. Wongsuphasawat, J.A. Guerra Gómez, C. Plaisant, T.D. Wang, M. Taieb-Maimon, and B. Shneiderman, LifeFlow: visualizing an overview of event sequences, in: ACM Press, 2011: p. 1747. doi:10.1145/1978942.1979196.

[23] J. Zhao, P. Papapetrou, L. Asker, and H. Boström, Learning from heterogeneous temporal data in electronic health records, J. Biomed. Inform. 65 (2017) 105–119. doi:10.1016/j.jbi.2016.11.006.

[24] M. Vandromme, J. Jacques, J. Taillard, A. Hansske, L. Jourdan, and C. Dhaenens, Extraction and optimization of classification rules for temporal sequences: Application to hospital data, Knowl.-Based Syst. 122 (2017) 148–158. doi:10.1016/j.knosys.2017.02.001.

[25] C. Plaisant, Visualization of temporal patterns in patient record data, Fundam. Clin. Pharmacol. 32 (2018) 85–87. doi:10.1111/fcp.12322.

[26] R Development Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2011. http://www.R-project.org/ (accessed August 12, 2019).

[27] A. Khwaja, KDIGO clinical practice guidelines for acute kidney injury, Nephron Clin. Pract. 120 (2012) c179-184. doi:10.1159/000339789.