

UC Irvine

UC Irvine Previously Published Works

Title

CHCi - A Dynamic Data Platform for Clinical Data Capture and Use.

Permalink

<https://escholarship.org/uc/item/40r3w6kq>

Authors

Wu, Danny TY
Zheng, Kai
Bradley, David J

Publication Date

2018

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

CHCi – A Dynamic Data Platform for Clinical Data Capture and Use

Danny T.Y. Wu, PhD, MS^{1,2,3} Kai Zheng, PhD^{4,1,5}, David J. Bradley, MD²
¹School of Information, University of Michigan, Ann Arbor, MI; ²Division of Pediatric Cardiology, University of Michigan, Ann Arbor, MI; ³Department of Biomedical Informatics, University of Cincinnati, Cincinnati, OH; ⁴Department of Health Management and Policy, University of Michigan, Ann Arbor, MI; ⁵Department of Informatics, University of California, Irvine, CA

Abstract

All academic medical centers have a strong desire to maximize the value of their clinical data for secondary use purposes such as quality improvement (QI) and research. However, this need has not been adequately fulfilled due in part to the fact that the data capture functions in current electronic health record systems predominantly focus on clinical documentation and billing, lacking the flexibility to allow the collection of additional data elements critical to QI or research. To address this gap, we designed and developed a dynamic data platform to support clinicians' varied needs for recording additional data about their patients outside of direct patient care (e.g. classifying patient conditions based on the inclusion criteria of a research-oriented patient registry). In this paper, we describe the design considerations of this platform such as data models, query functions, coding and controlled vocabulary, user interface design, access control, and data interoperability. In developing the platform, we partnered with the frontline clinicians in an academic congenital heart center, and adopted the agile software development approach with numerous rounds of evaluation and iterative refinement. Since 2013, this platform has been successfully used to meet the dynamic QI and research data needs of clinicians in the congenital heart center. Future work includes improving the efficiency and effectiveness of the platform and incorporating cutting-edge data interoperability standards.

Introduction

Academic medical centers are dedicated to providing high-quality patient care and conducting clinical and translational research (1,2). To achieve these goals, it is essential to ensure accurate and comprehensive recording of clinical data. For example, without such data, the promise of genome-wide association studies and comparative effectiveness research cannot be adequately fulfilled. This critical need, however, has not been fully met by current electronic health record (EHR) systems, as the priority in the design of these systems is to support direct patient care as well as the administrative processes (e.g., billing). They do not provide much flexibility to allow for the collection of extra data elements that are crucial for secondary use purposes such as quality improvement (QI) and research.

To address this gap, there is a need to extend current EHR systems' capability so that clinicians can, as part of their patient care workflow, record additional information about their patients or clinical processes that can be later used to facilitate QI and research (3,4). Currently, there have been initiatives that utilize application programming interfaces (APIs) or other means of integration to build research-oriented data collection into EHRs (e.g. integration with clinical trial management systems to support point-of-care patient recruitment). However, these solutions may be costly, cumbersome (e.g. 5), require a significant amount of vendor cooperations, and are difficult to maintain due to frequent vendor software upgrades and/or legal regulatory compliance. Therefore, these solutions do not fully address the changing data needs for supporting QI or research.

A potential solution is to develop companion software systems that work side-by-side with an EHR to provide supplementary functionality in supporting research data capture. There has been considerable effort in developing and adopting such IT solutions in healthcare institutions. However, existing solutions are either designed as standalone tools and are therefore difficult to integrate into clinicians' workflow, or are largely redundant with EHRs, requiring a significant amount of double data entry.

In this paper, we describe a dynamic clinical data platform that we designed and developed to address this gap. Prior to deploying this data platform, many database applications, using technologies such as Access databases, existed at a leading academic congenital heart center to support clinicians' data needs. However, these legacy database applications were implemented in an ad-hoc manner, resulting in common health IT problems such as duplicate programming effort and silo datasets. This was the main motivation for us to develop a unified platform that are highly modularized and customizable, and able to integrate discrete datasets together to facilitate downstream data uses to maximize the value of clinical data. This dynamic data platform has four design objectives: 1) high extensibility to

address clinicians' dynamic data needs in a timely fashion, 2) high accessibility to the data stored on the platform, whether structured or unstructured, 3) high accountability to justify the use of data records through detailed audit trail logs, and 4) high interoperability with other health IT applications, especially the EHR, to facilitate data sharing and minimize redundant data entry. In this manuscript, we report the design considerations of the platform, and discuss the value of this dynamic data platform through empirical evidence derived from the implementation of the system in a leading academic congenital heart center.

Methods

Clinical Setting

The design and implementation of the dynamic data platform were made possible through the collaboration of the Michigan Congenital Heart Center (MCHC), a multi-specialty inpatient and outpatient cardiac center where pediatric cardiologists, pediatric cardiac surgeons, certified nurses, and technicians work together to treat patients with pediatric and congenital heart diseases. Due to its size and multi-specialty nature, the need for clinical data capture and use in the center has become a more imperative issue than other clinical settings. For example, the MCHC catheterization team used a commercial case management tool to support its procedural workflow. This tool, however, shared no information with the institutional EHR system or with other MCHC teams, although it contained a potentially valuable dataset that could have been better utilized for quality improvement (e.g. complications review) and clinical research.

System Design and Evaluation

The implementation of the data platform in MCHC is named the "Congenital Heart Center intelligent and innovative data platform" (referred to as "CHCi" hereafter), which began in mid 2013. The database team has been working in the clinical setting since 2011, accumulating two years of experience prior to the implementation of CHCi. In these two years, the database team adopted traditional software development method (waterfall model) and standard relational database and web development techniques, producing four separate functional database applications to support four MCHC subgroups. A formal evaluation was conducted by a group of usability experts in the beginning of 2013 to understand the effectiveness, usability, and user acceptance of the four database applications, using human-computer interaction system evaluation methods including interactive map, comparative analysis, interview, survey, heuristic evaluation, and usability testing. The evaluation results suggest improvements on entry error prevention, navigation support, user interface design, and search and sorting functions. This formal evaluation led to the redesign of the software architecture and the adoption of agile software development method to create CHCi in mid 2013. Since then, the database team has been iteratively refining CHCi to enhance its ability to support the missions of many subgroups in MCHC using both top-down and bottom-up approaches.

After three years of use, CHCi has achieved many successes as described in our previous work (6). The current paper aims to share the design considerations and implementation challenges of this platform. One key design challenge was that each MCHC subgroup has both shared and unique data needs. CHCi was therefore designed to host multiple database applications for each of the subgroups. These related database applications were termed "modules" of the platform. The platform and its modules were designed such that a function of one module can be shared with, or easily reused by, another module. Similarly, the datasets captured in one module can be easily connected to the databases underlying the other modules. In the following section, we describe the approaches to address this design challenge and share technical details of CHCi with regard to the data model, controlled diagnostic codes, data search, user interface design, platform interoperability, and data access control. In terms of evaluation, we have demonstrated several successful case studies and preliminary evaluation results in our previous publications (7–9). In this paper, we provide another evidence to show the high extensibility of CHCi, which is measured by how many new modules were created and used between 2013 and 2016 versus the four data base applications implemented in the previous two years. We also provide informal evaluation evidence to show the high accessibility, accountability, and interoperability of CHCi. These evaluation results are described at the end of the results section.

Results

Data Model

An Entity-Attribute-Value (EAV) data model was chosen to support the storage and access of heterogeneous data on the CHCi platform (10). Table 1 shows an example of a traditional table and its corresponding EAV model. In this example, event attributes including event status and doctor of record in the left table as separate columns are transformed into key-value pairs in the right table. This decision was made because most current health IT systems are still supported by relational databases. Deploying an EAV model within a relational database, rather than using

schema-less databases, could lower the barriers to integrating platform data with existing data tables, resulting in higher interoperability of the platform. Utilizing such EAV data model can also favorably impact our two other design objectives: 1) high extensibility: adding an attribute does not change the table schema and the corresponding query functions; 2) high accessibility: data records can be easily queried against a smaller set of columns with limited or no effort to join multiple tables to obtain information (11).

Table 1. An example of a traditional table (left) and its corresponding Entity-Attribute-Value (EAV) data model (right)

MRN*	CASE_ID	STATUS	DOCTOR
0001	01	Admitted	Bradley
0001	02	Admitted	Wu
0002	03	Discharged	Bradley

* Medical Record Number

SEQ	MRN	CASE_ID	ATTRIBUTE	VALUE
1	0001	01	STATUS	Admitted
2	0001	01	DOCTOR	Bradley
3	0001	02	STATUS	Admitted
4	0001	02	DOCTOR	Wu
5	0002	03	STATUS	Discharged
6	0002	03	DOCTOR	Bradley

Moreover, the EAV model in our platform is expanded to achieve high accountability and modularization. With additional user identifiers and timestamps, our EAV model keeps all the “transactions” of data entries and can effortlessly generate audit trail logs and data entry trajectories, leading to higher data accountability since no data records are deleted on the platform.

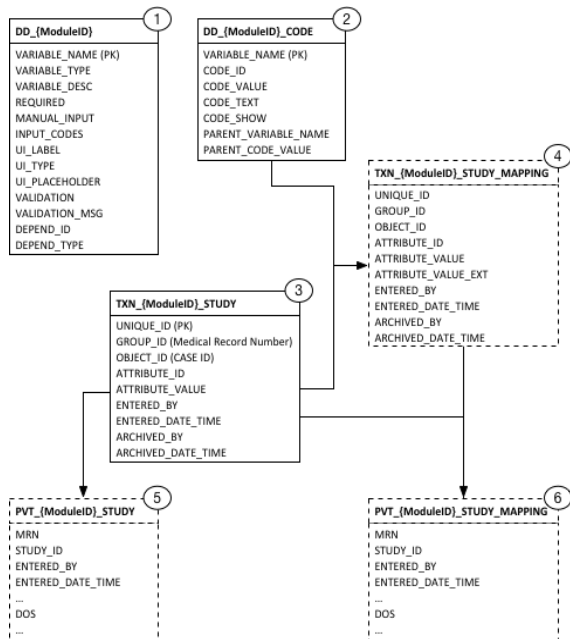


Figure 1 – CHCi EAV Data Model (the links indicate the view generation process, not table relationships).

In terms of modularization, each module on the platform is supported by exactly the same six tables, including the EAV model and five supporting tables and views. As shown in Figure 1, a module contains: 1) a data dictionary storing user-defined variables, 2) a code table storing discrete options for the variables, 3) a transaction table (EAV model) that keeps all the data entries, 4) a view combining the transaction table with the code table to facilitate data searches, and 5,6) two additional views to “pivot” data in the EAV model to a traditional table schema to support data access and downstream uses. These pivot tables are necessary because EAV models are not very human-readable and cannot be easily manipulated or combined with other data for analytics. Having the same table schema across all modules unifies the module creation process on our data platform. Moreover, this model shortens the development cycle by reusing the same data manipulation functions (e.g. UPDATE and ARCHIVE) across all modules. It is worth noting that all the values stored in our EAV model are converted into strings no matter what their original data type is. The original data types are specified separately in the data dictionary. This design allows all data entry records to be stored in one EAV model to simplify data storing and manipulating processes.

Controlled Diagnostic Codes

In the data model, all data variables can be flexibly defined by module users except diagnostic codes due to the unique value provided by diagnostic coding in clinical work and research. A special diagnostic code table is created and maintained by the database team, in particular under a centralized control of the lead physician. Module users, however, can create a temporary code if they feel the current code book is lacking, along with a code request to the lead physician. These code requests are carefully reviewed by the lead physician and the decision is made either to creating a new code, modify an existing one, or assign an existing code to the requested term. Temporary codes are automatically converted into permanent ones based on these decisions. This design has the benefit of ensuring the uniformity and usefulness of the diagnostic codes on our platform.

Another design consideration related to diagnostic codes is the adoption of Interface Terminology, the idea of using locally agreed terms by clinicians to facilitate clinical data entry, rather than using standardized terminology such as SNOMED CT or ICD9 codes (12). To promote downstream data uses, interface terminology can be later converted to standardized terminology. Figure 2 shows an example of our diagnostic code component shared across all CHCi modules. In this example, a term “down” is searched for “Down’s Syndrome” and a list of possible diagnostic codes is suggested by the data platform. The searcher picks the code “14.01.02” with a formal description “Trisomy 21”, which is automatically mapped to standardized terminology such as the International Paediatric Congenial Cardiac Code (IPCCC) and the Society for Thoracic Surgeons (STS) nomenclature frequently used in Pediatric Cardiology.

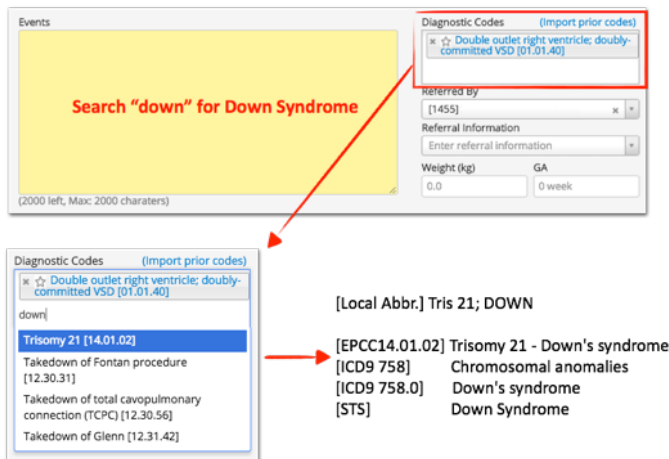


Figure 2 – Searching “Down’s Syndrome” [top right] in the Diagnostic Code Component (the results rank “Trisomy 21” the top choice [bottom left], which is mapped to other standard codes [bottom right])

Data Search

Our data model accommodates both structured and unstructured clinical data. Here, unstructured data refer to free-text clinical documents such as progress notes and discharge summaries. These notes are exported from our institutional EHR system’s data repository as plain text files and their note identifier is stored in the transaction table (EAV data module) as a key-value pair. The text files are further indexed and maintained by Apache Lucene (a full-featured text search engine library) to allow basic information retrieval. Figure 3 shows the general search process of the platform. Take the search term “smoker” for example, this keyword is first queried against the EAV data model and its views to identify records with “smoker” as part of the title and the value string. Next, this keyword is searched against the text indexes to retrieve notes with matched keywords. Then, these two results are combined at the case level, and eligible patient cases are presented with matched variables shown in context.

By combining the records in our EAV model and the indexes in Apache Lucene, this design can fully utilize the benefits of both structure and unstructured data and therefore decreases the tension between these two data types. As Rosenbloom et al. indicate, this tension comes from the complementary nature of the two data types. While structured data have the benefits of standardization and computer analyzable format, unstructured data have higher expressivity, flexibility, and may be more suitable for workflow (13). The ability of our data platform to search both structured and unstructured data can lower this tension and increase the utilization of clinical data regardless of their data types.

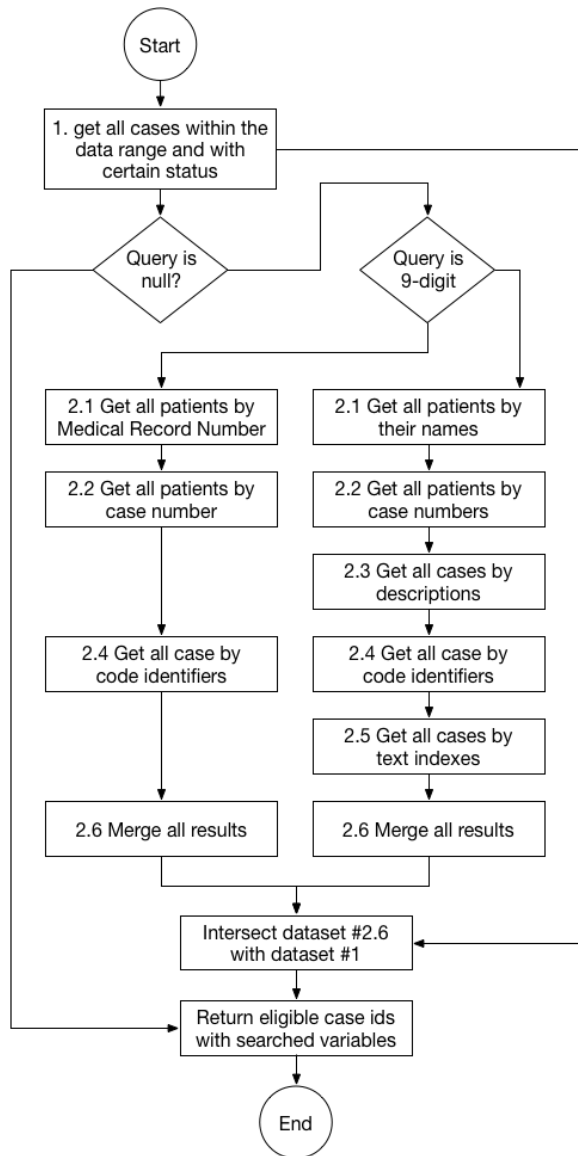


Figure 3 – General Search Process in CHCi

User Interface Design

The user interface is separated in two levels to help module users navigate through the data to accomplish their tasks. At the first level, the interface is designed to provide an “overview” of case events along with various ways to access these events and to export them. This interface consists of a landing page listing patient cases in a reverse chronological order by default (Figure 4). Module users can then browse case events using the paging options, search cases by entering keywords in the general search bar, filter cases based on predefined values in the advance search panel, and save search results in the CSV format by clicking the export button. At the second level, the user interface is designed to support data entry and verification. This second-level edit page shows patient demographics on top with additional information (e.g. gender, date of birth, and age) in the same order as the inpatient wrist band, followed by all cases of this patient to form a medical history (trajectory) so that a case can be interpreted in the context of adjacent ones (Figure 5). All the data entry activities on this edit page are automatically recorded using asynchronous JavaScripts (jQuery) to prevent data loss in a busy clinical environment. To facilitate case browsing, the content of a case can be collapsed and expanded by clicking its header. In addition, audit trail logs of data entries can be easily reviewed by clicking the “Hx” button to the right of the case header. This can strengthen module users’ trust in data integrity of the platform.

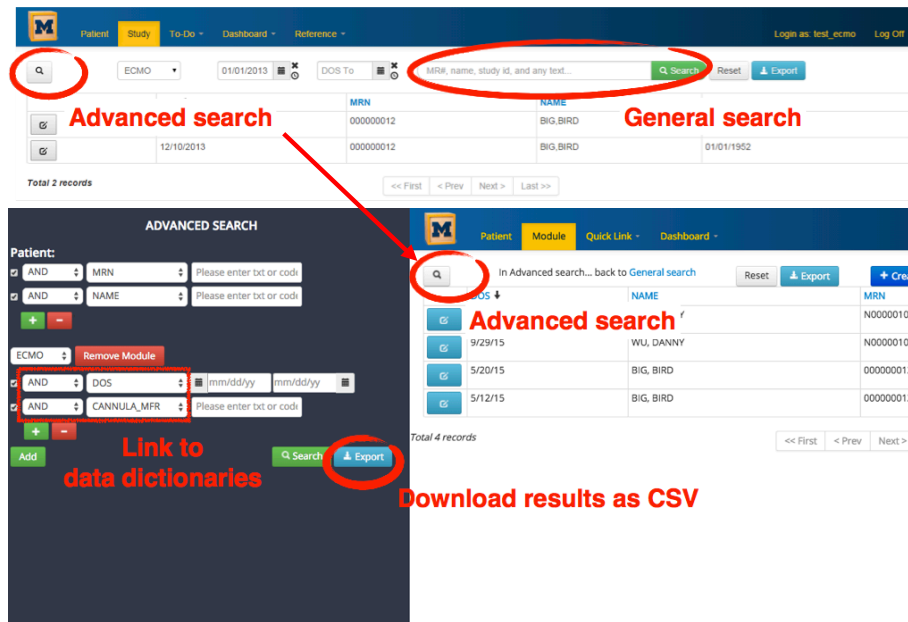


Figure 4 – First-level landing page of ECMO module (top: general search, bottom: advanced search)

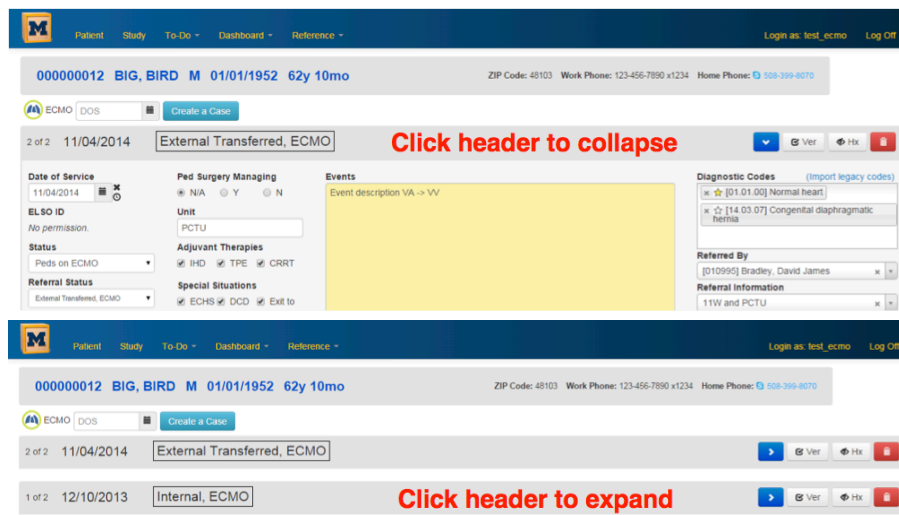


Figure 5 – Second-level edit page of ECMO Module (top: patient demographics and the most recent case, bottom: a list of cases with expandable/collapsible content to form patient medical history)

This two-level user interface design nicely balances the need for uniformity and customization of the CHCi modules. Since the first-level landing page can be shared across modules, it maintains the same look-and-feel and minimizes implementation effort when creating a new module. This landing page can also be configured to show different columns and identifying icons for each module through a database table. On the other hand, the second-level edit page is highly customizable. It can present data elements in various layout components (e.g. dropdown, radio button, text box, etc) and equip them with interactive features (hide/show, auto-calculation, data validation, etc). This high customizability results from the modularization of user interface components and the linking between these interface components and the data dictionaries. Taking dropdown creation for example, a variable is first entered in the data dictionary with its options being listed in the corresponding code table. Then, a dropdown interface component (programming function) is called, with parameters indicating the variable in the data dictionary to create a usable dropdown on the edit page. This dropdown can be easily repositioned on the edit page based on users' preferences. If

the dropdown needs complicated interactive features, these features can be implemented in JavaScript as a parameter of the interface component.

Platform Interoperability

The data platform currently has three major data interfaces, namely 1) web services, 2) direct database connections, and 3) file transfer. The choice of data interfaces depends on the target data sources and the timeliness of data needs. Figure 6 illustrates the conceptual model of the platform interoperability. For connections to the institution's EHR system, the platform employs XML-based web services if real-time data access is necessary. Otherwise, the platform retrieves data from the EHR data repository through a direct database link. For connections to other health IT applications, the data platform either uses a direct database link if permitted, or receives data files from those applications. These connecting processes are automated and run in batches in various intervals, between daily to monthly, depending on the module needs. Once the data platform collects all desired clinical data from heterogeneous sources, module users can interact with them through the user interfaces as described in the previous section.

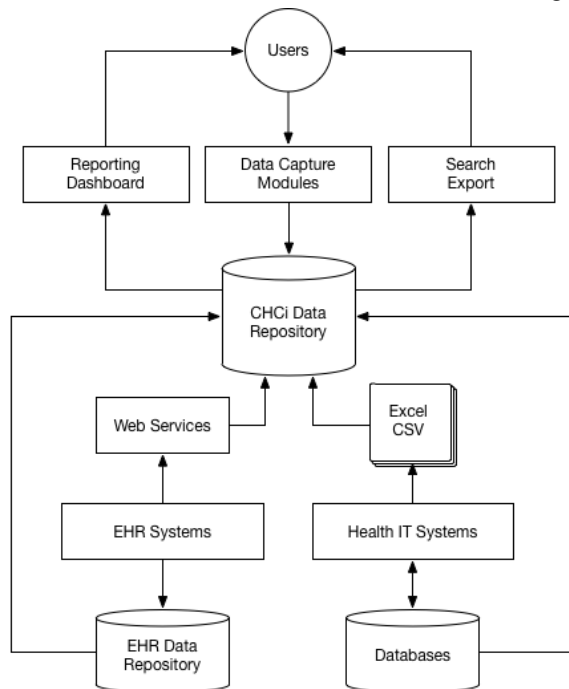


Figure 6 – Conceptual Model of CHCi Interoperability

Data access control

The data platform consists of a web server and a database server maintained by the University of Michigan Medical School Information Technology team. These two servers are hosted behind the Medical School firewall and cannot be accessed outside of the network without using Virtual Private Network (VPN). Even with a VPN and Medical School credentials, users must be granted access to the platform to link their credential to the platform login. Moreover, platform users can be granted different levels of access depending on their needs and job responsibilities. The types of access include 1) read-only, 2) read and write, and 3) full (read-write-verify) access. Data access is module specific, meaning that a user can be assigned read-only access to one module and full access to another. In addition to accessing data through the user interface, users can submit data requests to the database team. However, each research-oriented data request must have its own approval from the Institutional Review Board. Data requests are carefully reviewed and approved by the lead physician before being executed by the database team.

Implementation

The dynamic data platform (CHCi) was developed between January and June 2013, and has been running in the congenital heart center since September 2013. The platform is implemented in ASP.NET MVC 4.5 and hosted on a Microsoft IIS 7.5 web server. MVC (Model-View-Controller) is a software design and development framework that separates data from their business logic and representation to create a cleaner software architecture. The extended

EAV data model is implemented in an Oracle Database 12c Enterprise Edition 64-bit. Clinical notes are stored as plain text files and indexed using Lucene.Net 3.0.3. The user interfaces are programmed using HTML5 and JavaScript libraries including JQuery 2.0 and Twitter Bootstrap 2.3.2.

Evaluation

The high extensibility of the dynamic data platform is supported by the fact that the database team has been able to fast prototype new modules and iteratively refine them. Prior to using this data platform, between July 2011 and August 2013 (26 months), the database team deployed only four modules in an ad-hoc manner for the magnetic resonance imaging (MRI), neurodevelopmental follow-up (ND), ambulatory cardiac recording (HOLTER), and fetal cardiology (FETAL) teams to support their clinical work and research needs. After using CHCi, between September 2013 and December 2016 (40 months), the database team not only migrated the four legacy modules to CHCi, but also added nine more fully functional modules. Table 1 lists all CHCi modules as of 2016, their status, and the number of users. The total number of users is not the sum of all module users because one user may play multiple roles in different modules and can have a different level of access.

Table 1 – Current modules on the dynamic data platform

Module ID	Description	Status*	# Users**
MRI	Magnetic resonance imaging	M	2
ND	Neurodevelopmental follow-up	M	3
HOLTER	Holter (ambulatory electrocardiography)	M	8
FETAL	Fetal cardiology	M	4
CATH	Cardiac catheterization	M	5
LONG	Longitudinal follow-up	M	4
PC4	Pediatric Cardiac Critical Care Consortium – Intensive care	M	4
ECMO	Extracorporeal membrane oxygenation	U	7
ECHO	Echocardiography	U	4
NOTE	Clinical notes	U	10
ACHD	Adult congenital heart Disease	U	5
EXER	Exercise Laboratory	D	8
OHT	Transplant service	D	3

* M (Maintenance), U (User Testing), D (Development)

** Physicians, nurses, technicians, project coordinators, and data analysts

The high accessibility of the platform is attested by the frequent user-generated search and data export on the platform. For example, users exported about 202 and 174 datasets in calendar years 2015 and 2016, respectively, in addition to ad-hoc data requests that directly pull records from the database. While the high accountability of the platform is not documented systematically, the database team did occasionally encounter situations where module users needed to revert the current value of a variable to a previously entered value. Since our EAV model does not delete any records, this type of request is very easy to handle, which increases the accountability of the platform. Lastly, the interoperability of the platform, especially with the institution's EHR system, is highly appreciated by users. For example, based on our pilot evaluation, bringing all information to a centralized place for the Holter users as a result of high interoperability likely contributed to the substantial improvement in case turnaround time (7).

Although this implementation has shown the achievement of our four design objectives, it also faced a potential scalability challenge. That is, when a module had too many records, it did not respond to a search query very quickly. For example, 142,000 historical ECHO cases with 12 variables of each were imported to our platform, leading to 1.75 million records in the EAV data model. Querying these pivoted tables to complete data search became slow. The root cause of this issue had been identified, including: 1) the suboptimal design of the SQL statement of the pivoted views, 2) insufficient indexes on the clinical notes, and 3) inefficiency of the query process. Two solutions have been implemented to resolve this scalability issue. First, to enhance the EVA data model, the CHCi modules with a large amount of records have extra physical data tables with exactly the same table schema as the pivoted views to avoid suboptimal queries. A set of functions have been implemented to ensure consistency between the data stored in the EAV data model and these new physical data tables. Second, the indexes of clinical notes were rebuilt to include more metadata of the notes to support record filtering in the query process. These two solutions have successfully shortened the search time in the ECHO module, and have been applied to other modules facing similar challenges to improve the overall scalability of the platform.

Discussion

We designed and developed a dynamic clinical data platform (CHCi) with high extensibility, accessibility, accountability, and interoperability. Since 2013, CHCi has been successfully supporting clinicians' variety of data needs in the congenital heart center. Our data platform can collect and harmonize clinical data that were otherwise scattered and difficult to utilize. Here, we share three lessons learned in this journey. First, modularization is the key to our success. Since the data platform is highly modularized in both front-end (i.e. user interface) and back-end (i.e. data models), the database team can rapidly respond to clinicians data needs (i.e. fast prototyping) so that more resources can be devoted to improving data collection workflow, increasing user satisfaction, and conducting evaluation studies to ensure the effectiveness and efficiency of the platform (7–9). Also, the high modularization nature allows the database team members to cover each other's work when necessary and therefore makes it much easier to maintain the data platform. The second lesson we learned is the necessity of iteratively improving our modules since users' data needs are varied and may evolve overtime. The database team works closely with the end users to ensure that the platform meets their data needs and to identify future improvement opportunities. The third lesson is that all stakeholders have come to appreciate that a significant amount of effort must be applied to educating the user base about new developments. This strengthens the trust among existing users and make the adoption easier for prospective users. It also garners essential organizational support to sustain this important work.

Through the implementation of this dynamic data platform, we have identified a potential scalability issue in the design and had developed two solutions to address this issue as described above. We will continue improving the dynamic data platform to ensure the achievement of our four design goals and the overall efficiency and effectiveness of the platform. Meanwhile, we are enhancing the platform's ability to support clinical research. For example, we have started to work with the research group in MCHC to develop a patient cohort management tool to facilitate the identification of study populations based on the case attributes and notes. In addition, we would like to increase the portability of the data platform to benefit other health institutions facing similar data capture issues. We have started to re-engineer the next version of the platform, automate the deployment processes, and test cutting-edge data interoperability standards, such as SMART-on-FHIR (14). We are very excited to continue this journey to develop this dynamic data platform to support clinical data capture and facilitate data use, and demonstrate the value of our platform in reducing clinicians' data entry burdens, improving care quality, and supporting long-term registry management.

Conclusions

In this paper, we present the design considerations of our dynamic data platform, which has been implemented at an academic congenital heart center and successfully supports the clinicians' dynamic data needs since 2013. We learned that modularization, iterative improvement, and user education are the keys to our success. We believe the platform has great potential to serve as a long-term registry management system, and continue improving the platform, enhancing its ability to discover patient cohorts, and incorporating cutting-edge data interoperability standards.

Acknowledgments

We thank Dr. Caren Goldberg, Dr. Sara Pasquali, Dr. Gerold Serwer, Mr. Ray Lowery, Ms. Janet Donohue at the Michigan Congenital Heart Center for their advice on improving the usability and usefulness of the platform. We also thank Ms. Yue (Anna) Yu for her help summarizing the achievements of the platform. This project was not supported by any grant at the time of submission. The first author worked part-time as a senior database analyst/programmer at

the Michigan Congenital Heart Center between 2011 and 2016 when he was pursuing his PhD degree at the University of Michigan School of Information, and served as a technical consultant to the project between March and October 2017 after graduation.

References

1. Pinsky WW. The roles of research in an academic medical center. *Ochsner J*. 2000 Oct;2(4):201–2.
2. U.S. Food & Drug Administration. Clinical Research Versus Medical Treatment [Internet]. 2017. Available from: <https://www.fda.gov/forpatients/clinicaltrials/clinicalvsmedical/default.htm>
3. Ahmad A. Electronic health records - beyond meaningful use. *Appl Clin Inform*. 2010;1(3):265–7.
4. Hersh WR, Cimino J, Payne PRO, Embi P, Logan J, Weiner M, et al. Recommendations for the Use of Operational Electronic Health Record Data in Comparative Effectiveness Research. *EGEMs Gener Evid Methods Improve Patient Outcomes* [Internet]. 2013 Oct 8 [cited 2014 Jan 1];1(1). Available from: <http://repository.academyhealth.org/egems/vol1/iss1/14>
5. Murphy EC, Ferris FL, O'Donnell WR. An electronic medical records system for clinical research and the EMR EDC interface. *Invest Ophthalmol Vis Sci*. 2007 Oct;48(10):4383–9.
6. Bradley DJ, Wu DTY, Goldberg CS, Serwer GS, Lowery RE, Donohue JE, et al. Out of many, one: integrating data in the paediatric cardiovascular environment. *Cardiol Young*. 2016 Sep 29;1–7.
7. Wu DTY, Zheng K. Implementation of a Computer-Based Documentation System Improves Workflow Efficiency: A Case Report. *American Medical Informatics Association*. DC.; 2014.
8. Wu DTY, Lowery R, Bradley DJ. Developing an EHR-based Approach to Identify Pediatric Single Ventricle Patients. *American Academy of Pediatrics*. Chicago, IL. COCIT Abstract.; 2017.
9. Wu DTY, Bradley DJ. Improving the quality of the Michigan fetal heart program using a dynamic data platform. *American Academy of Pediatrics*. Chicago, IL. COCIT Abstract.; 2017.
10. Nadkarni PM, Marenco L, Chen R, Skoufos E, Shepherd G, Miller P. Organization of heterogeneous scientific data using the EAV/CR representation. *J Am Med Inform Assoc JAMIA*. 1999 Dec;6(6):478–93.
11. Chen RS, Nadkarni P, Marenco L, Levin F, Erdos J, Miller PL. Exploring performance issues for a clinical database organized using an entity-attribute-value representation. *J Am Med Inform Assoc JAMIA*. 2000 Oct;7(5):475–87.
12. Rosenbloom ST, Miller RA, Adams P, Madani S, Khan N, Shultz EK. Implementing an interface terminology for structured clinical documentation. *J Am Med Inform Assoc JAMIA*. 2013 Jun;20(e1):e178-182.
13. Rosenbloom ST, Denny JC, Xu H, Lorenzi N, Stead WW, Johnson KB. Data from clinical notes: a perspective on the tension between structure and flexible documentation. *J Am Med Inform Assoc*. 2011 Mar 1;18(2):181–6.
14. Mandel JC, Kreda DA, Mandl KD, Kohane IS, Ramoni RB. SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. *J Am Med Inform Assoc JAMIA*. 2016 Sep;23(5):899–908.