The Development of a Medication Reconciliation Application

October 13, 2020

Report prepared for:
The State of Connecticut Office of Health Strategy

Report prepared by:
Thomas P. Agresta, MD, MBI, UConn Health
Steven A. Demurjian, PhD, UConn
Eugene Sanzi, PhD, UConn
Ryan J. Tran, MS, UConn Health
Rachel S. Rusnak, MPA, UConn Health

Table Of Contents

Introduction

In the United States medications are prescribed as first line treatment for 88% of chronic diseases[1]. In 2014 the Centers for Disease Control and Prevention (CDC) reported that 40.7% of seniors (65 years or older) and 10.9% of the total population were taking five (5) or more prescriptions. For seniors, the 40.7% represents almost a three-fold increase from the period of 1988-1994 (13.8%)[2]. Since a patient's medication regimen is the basis for treatment decisions, it is critical that medication lists are accurate to maximize therapeutic impact and prevent potentially life-threatening events. Despite this fact, there are often major discrepancies between the medication lists within various HIT systems where patients receive care, leading to the potential for substantial patient harm.

In an effort to make progress in addressing challenges around polypharmacy the medication reconciliation application development project focused on three specific problems facing physicians during medication reconciliation: obtaining medication list data from multiple disparate sources (e.g., EHRs from multiple providers); merging the complete medication list data at one, easy to access location; and, processing the complete medication list data to provide a Best Possible Medication History (BPMH). This BPMH could then be displayed to a physician, allowing the physician the ability to confirm that the medications presented are correct. On this premise the project team worked to develop an interface utilizing the Fast Healthcare Interoperability Resources (FHIR) standard to pull medication records from multiple electronic data sources and perform medication reconciliation, delivering a complete, merged list of medications.

Please note that additional details on the work presented in this report can be found in the peer reviewed paper that was accepted and will be presented at a conference in October 2020: Demurjian, S., Agresta, T., Sanzi, E., DeStefano, J., Ward-Charlerie, S., Rusnak, R., and Tran, R., "A Mobile Health Application for Medication Reconciliation Using RxNorm and FHIR," Proceedings of the Fifth International Conference on Informatics and Assistive Technologies for Health-Care, Medical Support and Wellbeing, HEALTHINFO 2020, October 2020.

[1] M. Ekstrand, "Transforming "Med Wreck" into "Med Rec:" One Health System's Journey," Webinar presentation: Pharmacy Quality Alliance, July 2017.
[2] State of Connecticut, "An Act Requiring the Health Information Technology Officer to Establish A Working Group to Evaluate Issues Concerning Polypharmacy and Medication Reconciliation," State of Connecticut Senate, Hartford, 2018.

Background

In 2018, recognizing challenges associated with polypharmacy, the CT General Assembly passed Special Act 18-6: An Act Requiring the Health Information Technology Officer to Establish a Working Group to Evaluate Issues Concerning Polypharmacy and Medication Reconciliation[3]. They utilized the Joint Commission's[4] definition of Med Rec: "Medication reconciliation is the process of comparing a patient's medication orders to all of the medications that the patient has been taking. This reconciliation is done to avoid medication errors such as omissions, duplications, dosing errors, or drug interactions." The group convened in September 2018, and developed and delivered a set of 11 recommendations around Medication Reconciliation and Polypharmacy.

Meanwhile, in the Spring of 2019 UConn Health and OHS hosted a Medication Reconciliation Hackathon[5], attended by 39 clinicians and 40 technical individuals. The two-day hackathon demonstrated the way that current and emerging technology standards, such as the FHIR RESTful API and RxNorm could improve the acquisition of a medication list and permit new user interfaces and features (e.g. specialty applications or features in a patient portal that could empower the patient (or guardian/parent) to report useful information (e.g. side effects, adherence, and undocumented OTC meds, prescriptions and supplements)) and improve the longitudinal sharing of this information across platforms and venues of care.

While there are general hazards associated with lack of comprehensive medication lists, transitions of care (hospital to ambulatory care, specialist to primary care, nursing facility to home) present a particularly critical and dangerous period for patient care due to lack of an accurate medication list as one (or more) provider(s) hands off to another provider or care team. This is especially true when a different (or no) EHRs are used between care locations, with medication errors contributing to this harm.

A number of different technology solutions exist; however, they are limited by the completeness of the data gathered and the ability to interoperate with other HIT systems. Examples of existing technology include Surescripts Medication History which gathers electronic prescription data, but misses patient reported medications and non-electronically prescribed medications; Cureatr which allows a pharmacist to use a standalone product that doesn't allow consumers access and is not interoperable; and CareEverywhere from EPIC which is vendor specific and still keeps information siloed from where it might be needed most.

With medication errors accounting for over 1 million emergency visits, 3.5 million physician office visits and over 125,000 hospital admissions annually[6], this problem clearly persists. Effective medication reconciliation can improve all of these issues, by compiling and displaying a list of medications from various EHRs into a single source-of-truth which users would access seamlessly within their HIT systems for medication management.

---

[3] CT State Legislature, "Connecticut SB00217 | 2018 | General Assembly," LegiScan, 2018. https://legiscan.com/CT/bill/SB00217/2018.

[4] The Joint Commission, "Medication reconciliation sentinel event alert," 2006.

[5] https://health.uconn.edu/quantitative-medicine/hackathon-2019/

[6] Office of Disease Prevention and Health Promotion, "Adverse Drug Events," February 2020. https://health.gov/our-work/health-care-quality/adverse-drug-events.

The Project

Armed with the experience and recommendations of the Medication Reconciliation and Polypharmacy Workgroup, the results of the Hackathon, and insights from the Medication Reconciliation Polypharmacy Committee, the project team sought to demonstrate how automated medication reconciliation has the potential to reduce medical mistakes and improve patient outcomes.

A critical time for the patient is when the patient's care is transferred to another provider (Clinician or Visiting Nurse for example), as it is not guaranteed that each provider has a complete and accurate representation of the medications that the patient is taking. Medication reconciliation is required at each point of transfer of the patient's care, but reconciliation can be difficult for the healthcare provider due to: the spread of patient healthcare data among multiple EHRs (e.g., prescription created by the physician's EHR and sent to be fulfilled by the pharmacist's EHR); incomplete or non-existent interoperability between healthcare systems; incomplete or incorrect medication lists; and, the difficulty of displaying potentially conflicting medication data entries to the physician on a potentially small screen (e.g., smartphone, tablet, etc.).

The goal of the MedRec project is to develop an early prototypical design of a single point of consolidating the information in the form of an application programming interface (API) that a physician's computer device can access to obtain a BPMH. This API should be capable of retrieving medication lists from multiple EHR sources for a specific patient, with an easy method for adding, removing, or editing which sources the API should retrieve medication data from. The medication lists from each source should then be merged into one comprehensive medication listing, then processed to find potential duplicates or errors. The complete medication list and list of potential duplicates and errors is then sent to an app, which processes the data into a BPMH. This BPMH should then be displayable in an app, built for iOS and Android. The retrieval and processing of the complete medication list should occur completely transparently without the necessity of intervention on the physician's behalf. To a physician or other user, the retrieval and processing of the BPMH should appear no differently than if the medication list was pulled from their own, local EHR. The BPMH display should also highlight the lists of potential duplicates and errors to the physician indicating which medications may be erroneous or need to be merged into one entry. Once the physician has confirmed that the BPMH is accurate, the patient's BPMH should be saved to the MedRec server for later use.

The Process and Team

A multidisciplinary team met weekly for seven months with the goal of further refining the business and technical issues required to develop an APP to display the BPMH. A Ph.D. graduate student in Computer Science and Engineering iteratively designed, developed and deployed in a test environment, software applications that carried forward the emerging vision of the group. A number of substantial interoperability, ontology and semantic other issues were identified which were also passed off to the Medication Reconciliation and Polypharmacy Committee (MRPC) group in order to help with their development of a set of recommendations regarding next steps for the state of Connecticut and its designated HIE – CONNIE as they consider appropriate solutions for Medication Reconciliation.

Team Members

- **Thomas Agresta M.D., MBI.:** Professor and Director of Medical Informatics Family Medicine, Director of Clinical Informatics Center for Quantitative Medicine (CQM) University of Connecticut School of Medicine
- **Steven Demurjian Ph.D.**: Professor of Computer Science and Engineering (CSE) University of Connecticut
- **Eugene Sanzi Ph.D.**: Primary Software Developer while a Graduate student in CSE University of Connecticut
- **John DeStefano PharmD**: Pharmacy Informatician, SMC Partners Inc.
- **Ryan Tran MHS**: Research Assistant, CQM University of Connecticut School of Medicine
- **Rachel Rusnak MPA**: Research Associate, CQM University of Connecticut School of Medicine
- **Stacy Ward-Charlerie PharmD**: Pharmacy Informatics Consultant, WardRx

Accomplishments

The MedRec project provides an interface utilizing the Fast Healthcare Interoperability Resources (FHIR V4.01) standard in order to fulfill the Office of the National Coordinator of Health Information Technology's (ONC) requirements for interoperability under the 21st Century CURES Act[7]. The FHIR interface can be connected to by any client capable of interacting with and parsing the medication data returned by a FHIR interface. The FHIR interface can be configured to pull from multiple FHIR endpoints representing multiple, disparate EHRs. This specific API was chosen since it is the ONC required standard for interoperability that all hospitals and other users of Certified HIT technology must be able implement. The functional capability of the App was

[7] https://www.healthit.gov/cures/sites/default/files/cures/2020-03/APICertificationCriterion.pdf

tested by duplicating an instance of the medication data from the 2019 OHS / UConn Health sponsored - Medication Reconciliation Hackathon[8].

The architecture for the MedRec testbed is depicted in Figure 1. The MedRec Backend Server, depicted in the green box in the center of the figure, represents the MedRec FHIR endpoint that obtains medication data from other EHRs, merges the medications into one listing, performs medication reconciliation on the listing, then outputs the reconciled medication list in FHIR's JSON (Java Script Object Notation) format[9]. To the left of the MedRec Backend server are the NDC (National Drug Code) code[10], RxNorm[11], and RxTerms APIs[12] utilized by the MedRec Backend Server to match potential duplicates in the complete medication listing. The box to the right of the MedRec Backend Server is the OpenEMR[13] Docker[14] FHIR API, a collection of three Docker containers on a Docker network running an OpenEMR installation housing medication data from the Fall 2019 UConn Hackathon. Docker containers are similar to the concept of a virtual machine, they allow a developer to encapsulate components for an application (e.g., web server, a database server, and a functional server etc.) into one logical unit that is package for the release of a software system that can be easily installed and deployed with very few steps.

OpenEMR is an open source ONC certified electronic healthcare record system. The three Simulated EHR FHIR APIs located to the top of the image represent three EHRs with a permutation of the medication listings of the patients listed in the OpenEMR Docker FHIR API. When a physician requests medication data for a patient listed in the MedRec mHealth app, at the bottom of Figure 1, the MedRec mHealth app initiates a request to the MedRec Backend Server. The MedRec Backend server contacts the three Simulated EHRs and the OpenEMR Docker FHIR API, makes a request via the FHIR API for medication data belonging to the selected patient, merges all of the medication data returned by the three Simulated EHRs and OpenEMR Docker FHIR API into one FHIR Bundle resource, then processes all medications in the Bundle by attempting to match the medications through an locally developed algorithm that uses NDCs, RxNorm, and RxTerms. The results are sent to the MedRec mHealth App where the physician can view a reconciled medication list.

We describe the process in the following 4 subsections discussing the various parts of the MedRec testbed including: the MedRec Backend Server, the algorithm for medication reconciliation, the OpenEMR Docker FHIR API housing the Fall 2019 Hackathon data, and the MedRec app.

[8] Demurjian, S., Agresta, T., Sanzi, E., DeStefano, J., Ward-Charlerie, S., Rusnak, R., and Tran, R., "A Mobile Health Application for Medication Reconciliation Using RxNorm and FHIR," *Proceedings of the Fifth International Conference on Informatics and Assistive Technologies for Health-Care, Medical Support and Wellbeing,* HEALTHINFO 2020, October 2020.
[9] https://www.json.org/json-en.html
[10] https://www.fda.gov/drugs/drug-approvals-and-databases/national-drug-code-directory
[11] https://www.nlm.nih.gov/research/umls/rxnorm/index.html
[12] https://rxnav.nlm.nih.gov/RxTermsAPIs.html#
[13] https://www.open-emr.org/
[14] https://www.docker.com/resources/what-container

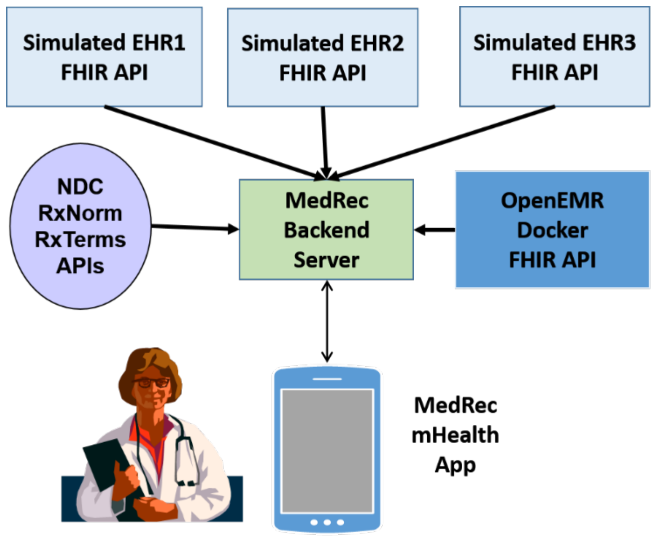

Figure 1. MedRec Architecture.

MedRec Backend Server

The MedRec Backend Server was written in Java utilizing the HAPI FHIR API, a Java implementation of the FHIR standard. The server is configured with a list of source URLs, each URL representing the FHIR endpoint the MedRec Backend Server should connect to through a FHIR API. The MedRec Backend Server can connect to multiple EHRs as shown in Figure 1, although presently only the OpenEMR Docker FHIR API is configured. The server makes client requests to each configured URL, retrieving a Bundle of FHIR Medication Statement resources and merging each Bundle into one Bundle containing all Medication Statements from all configured EHRs.

MedRec Algorithm

The MedRec Algorithm takes as input the complete, merged Medication Statement listing from all the sources (URLs) the MedRec Backend Server is configured to request medication lists from and creates a copy for processing. The algorithm begins by choosing the first medication. The medication's name is passed to the approximate search of the RxNorm API to retrieve a list of RxCUIs described by the medication name. For each RxCUI returned by the RxNorm API, a search is sent to RxNorm's all related API that returns a list of related concepts (e.g., drug synonyms, generic vs. brand name, ingredients, etc.). The related concepts are compared to the remaining medications in the complete medication list. If a match is found, the algorithm adds the pair of medications to a set of potential duplicates for later processing and removes the matched medication. Once the end of the list is reached, the first medication is removed and the process is restarted by attempting to match the second medication to each other medication in the list. Once there are no remaining medications to match, the set of potential duplicates is processed into a set of Detected Issues, a FHIR resource that denotes potential issues in returned data. The Detected Issues are then added to the original complete medication list Bundle and returned to the requestor (e.g., the MedRec app).

OpenEMR Docker FHIR API

The OpenEMR Docker FHIR API consists of a set of three Docker containers, collectively containing the Fall 2019 Hackathon data and the APIs necessary to access it. The first Docker container contains a HAPI FHIR implementation that can access a custom API we have integrated into OpenEMR. The first Docker container is the endpoint that a FHIR request is sent to. It retrieves the patient and medication data through the custom OpenEMR API and translates it into the FHIR standard before relaying it to the requestor. The second Docker container contains the OpenEMR installation and the custom data retrieval API. The OpenEMR installation acts as an easy method to facilitate the addition or editing of medication data for patients by healthcare professionals, while also representing the integration of a real-world ONC certified EHR system with our MedRec implementation. The third Docker container contains the MySQL database that OpenEMR uses to store its patient records. All three Docker containers are configurable through a provided .env file that specifies environment variables necessary to make the Docker containers accessible to each other and to the environment outside of the Docker network.

MedRec App

The MedRec app provides an example app capable of connecting to our custom MedRec FHIR server, making a request for patient data, and displaying the patient's reconciled medications. The MedRec app is written with Google's Flutter toolkit, which allows for the generation of both iOS and Android apps from one codebase. When the app is launched, it makes a request for a list of patients from the OpenEMR Docker FHIR API and displays the Patient List as shown in the left screen in Figure 2. When a patient is selected, the app shows a Patient Details screen as shown in the right screen in Figure 2. The Patient Details screen displays: the patient's name; the URL of the EHR that the Patient resource originated from, representing the patient; the patient's birthday; and, the patient's address. The bottom of the Patient Details screen contains two buttons: pressing Complete Med List shows all of the medications retrieved from all sources configured in the MedRec server, organized by the EHR they were retrieved from, as shown in the left screen of Figure 3; while pressing the Reconciled Med List will lead to a reconciled medication list as shown in the right screen of Figure 3. Note that presently the Reconciled Medication screen does not remove medications as in the right screen of Figure 3, but displays the Detected Issue resources in text form alongside the medications.

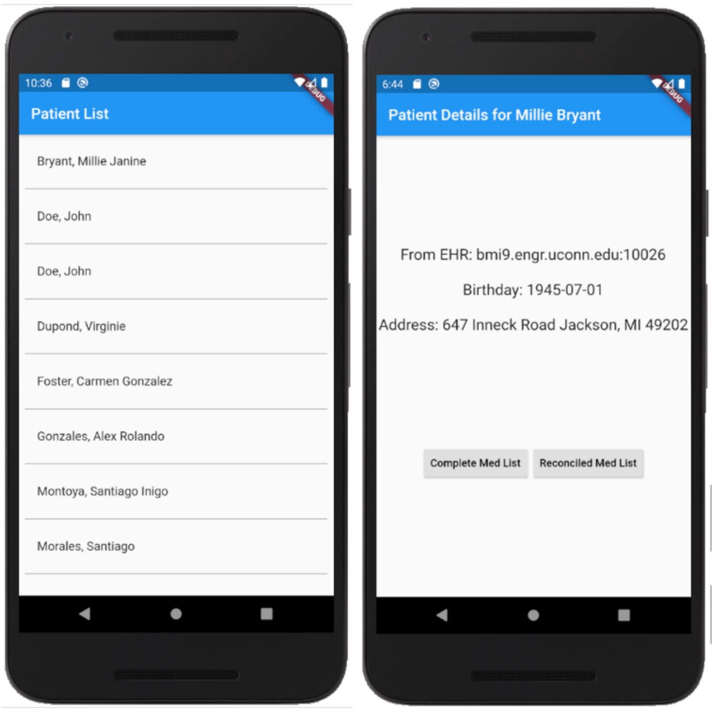

Figure 2. The Patient List and Patient Details Screens.

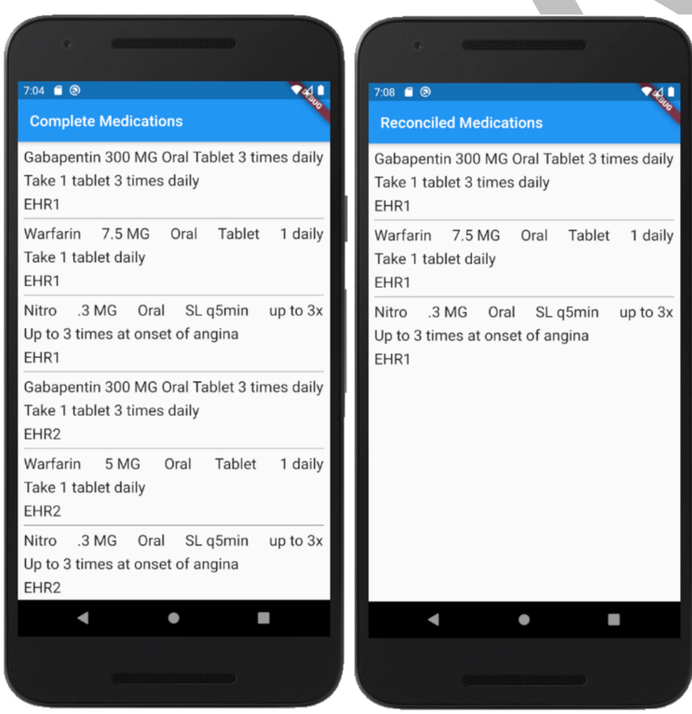

Figure 3. The Complete Medications and Reconciled Medications Screens.

Opportunity

Building on the progress made between January 2020 and August 2020 there is an opportunity to develop and test an end-to-end software solution designed to work at a state-level scale to create the Best Possible Medication History (BPMH). This could be accomplished by further developing a mobile health app for medication reconciliation suitable for use by patients and medical providers as well as an API based solution suitable for inclusion into electronic health records EHRs.

Recommendations to move forward would include utilizing the NCPDP Medication List Transaction to retrieve/collect medication information from EHR systems to electronically transmit it to and from a Health Information Exchange (HIE) for a potential state-wide process of determining BPMH within the state HIE[15]. This could allow multiple Health Information Technology (HIT) systems to share healthcare data, and improve patient treatment and satisfaction[16]. The team is pursuing opportunities to implement the Fast Healthcare Interoperability Resources (FHIR)[17] standard that the Office of National Coordinator for Health IT (ONC) approved to promote secure sharing of healthcare data between HIT/HIE systems as well as consumer facing healthcare applications[18] to prevent information blocking and promote interoperability.

In addition to planned future development of the application, focus groups should be convened to allow healthcare providers and or patient representatives the opportunity to interact with the software. This would allow providers to test the system with artificial patient data, and provide feedback on the interface and features. This feedback should be considered for incorporation of future iterations of this or another sanctioned system.

Updates to this work can be informed by the work of the Medication Reconciliation Polypharmacy Committee (MRPC) under the Health Information Technology Advisory Council for the state of Connecticut, which is finalizing business and technical recommendations for the BPMH[19]. Evaluation of these recommendations can provide guidance for next steps for future algorithm development, functional software access development as well as user interface development. In addition this product can be utilized by members of the MRPC and others to help iterate on rapid cycle design of this product as well as allowing a refinement of the BMPH business, functional and technical requirements.

---

[15] NCPD.org.
[16] HealthIT.gov
[17] HL7.org
[18] https://www.healthit.gov/isa/
[19] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5362254/

Considerations

Privacy & Security

"Health information privacy is an individual's right to control the acquisition, uses, or disclosures of his or her identifiable health data. Confidentiality, which is closely related, refers to the obligations of those who receive information to respect the privacy interests of those to whom the data relate. Security is altogether different. It refers to physical, technological, or administrative safeguards or tools used to protect identifiable health data from unwarranted access or disclosure."[20]

It is important to note that, the incorporation of any privacy or security controls was out of scope for this pilot project, but would be an integral part of an actual developed solution. This would include the need to ensure proper consent for usage and access to data was enforced, as well as the need for a highly secure, HIPAA compliant, HITRUST certified backend server infrastructure and front-end app were developed. It would be anticipated that these would be designed to meet the requirements of state and federal regulations as well as the CT State HIE requirements.[21]

Usability

Also out of scope for this pilot was end-user feedback for feature design and usability testing. These should be critical components of a robust engineering, development and implementation strategy for any software solution (or App development) intended to meet the Medication Reconciliation needs described by the MRPC on behalf of the HIT Advisory Council. A series of design sessions and usability lab evaluations would be part of a recommended next step for this or any other potential solution that is considered.

[20] http://www.ncvhs.hhs.gov/060622lt.htm.

[21] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4432854/

Conclusion

This small pilot project was able to provide evidence of the feasibility of developing a multi-component software solution that is able to pull (and potentially eventually push) medication data from a number of disparate sources, using a standardized and federally supported FHIR API and use additional data from federally maintained medication ontology services to normalize these lists, determine duplication and present to the end-user via a mobile app a consolidated medication list with observations about duplicity.

There were a large number of lessons learned during this process, which can be shared to facilitate effective next steps, as well as to inform future development of a solution to retrieve and normalize medication data, perform algorithmic actions upon it and provide access to end users for actionable use within their workflow.

We look forward to having the opportunity to have the work done and lessons learned to date, be integrated into Connecticut's quest to improve medication safety and healthcare delivery efficiency by sharing what has been accomplished during this project.

We are grateful to the support received from the Office of Health Strategy in Connecticut, through an MOA with UConn Health, to help provide some funding for this effort.

Appendices

Appendix A: Glossary of Technical Terms

FHIR (Fast Healthcare Interoperability Resources)

A specification, which is a standard for exchanging healthcare information electronically.
For more information visit HL7.org

RXNorm

RxNorm provides normalized names for clinical drugs and links its names to many of the drug vocabularies
For more information visit the National Library of Medicine

HIE

Electronic health information exchange (HIE) allows doctors, nurses, pharmacists, other health care providers and patients to appropriately access and securely share a patient's vital medical information electronically
For more information visit HealthIT.gov

BPMH (Best Possible Medication History)

Refers to the development of a comprehensive, accurate list of medications (prescription, over the counter, supplements, etc).
For more information visit the World Health Organization

NCPDP (National Council for Prescription Drug Programs)

Serves as a problem solving forum for healthcare, using consensus based standards development. For more information visit NCPD.org.

OPEN EMR

A free, open source electronic health record.
For more information visit OpenEMR.org.

DOCKER Container

A container is a standardized unit of software. A docker container is a lightweight, standalone, executable package of software that includes everything needed to run an application.
For more information visit docker.com.

JSON (Java Script Object Notation)

A lightweight data interchange format.
For more information visit json.org.

HAPI

A framework for building applications and services.
For more information visit hapi.dev

HapiFHIR

A complete implementation of the HL7 FHIR standard for healthcare interoperability in Java. For more information visit hapifhir.io

API (Application Programming Interface)

A software intermediary that allows two applications to talk to each other.

Appendix B: Next Steps in Development

The following activities are planned to further the work on the development of the medication reconciliation application.

- Review and incorporation of MRPC recommendations for medication reconcilliation
    - Review business and functional requirements from the MRPC regarding development of a BPMH and creation of technical requirements document from these recommendations.
    - Evaluation and gap analysis of current prototype to BMPH requirements
    - If appropriate development a roadmap to incorporate features of BMPH requirements
- Evaluation of options and opportunities for incorporation of current prototype designs and lessons learned from MRPC with regards to HIE Integration partners' capabilities and design pipeline
- Improvements to infrastructure:
    - Asynchronous FHIR requests to each source to improve data retrieval efficiency
    - Establishment of multiple test sources
    - Support for multiple access security types (e.g., username/password, certificate, OAuth, etc.)
    - Saving reconciled BPMH to the MedRec server
    - Improve Detected Issue utilization by using extensions to properly document the connections between the listed medications
    - Improvements to the reconciliation algorithm: efficiency and accuracy
        - Add time factor
        - Trace prescriptions from doctor to pharmacist to refills
        - Treat the medication listing and list of pairs of medications to be merged as a graph, with the medications forming nodes and the merged medication pairs as edges, allowing efficient path search algorithms to be utilized to connect medication entries
    - Move RxNorm usage from remote API calls to our own local RxNorm database
        - Integration of RxTerms
- Improvements to app:
    - Change from direct listing of the Detected Issue resource to a visual representation
    - Represent a timeline of the patient's medication history
    - Integrate the Dart FHIR library to prevent having to parse JSON manually
    - Allow physicians using the app to confirm/edit the BPMH
- Overall improvements to the system:
    - Utilize the feedback of the physicians as training data through supervised learning, allowing continuous improvement of the algorithm
        - Need to save the MedRec server's analysis to compare to the physician's final BPMH when it arrives
    - Method to combine physician-created BPMHs saved to the MedRec server with new medication entries that occurred after the reconciliation