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

doi:10.3233/SHTI200234

Integrating Pharmacogenetic Decision Support into a Clinical Information System

Kevin TIPPENHAUER[a,1], Marwin PHILIPS[a], Carlo R. LARGIADÈR[b], Murat SARIYAR[a], Thomas BÜRKLE[a]
[a]*Bern University of Applied Sciences,* [b]*Bern University Hospital (INSEL)*

Abstract. Pharmacogenetic testing can prevent adverse drug events but has rarely found its way into clinical routine. One reason is the lack of tools for smooth and automatable integration of pharmacogenetic knowledge into existing processes. Especially, electronic medical records (EMR) represent a suitable environment for such tools. We developed a modular service-oriented prototype of a pharmacogenetic decision support system within an EMR system of the Bern University Hospital. Here, we present the component architecture of our system and discuss issues required for generalizing our results.

Keywords. Pharmacogenomics, pharmacogenetics, decision support, adverse drug events, HL7 FHIR, expert system

1. Introduction

Pharmacogenetics is the study of genetic effects on the metabolic pathways of drugs. About 95% of individuals carry one or more known genetic variants that are important for drug dosing recommendations [1]. Ignoring genetic variation while prescribing a drug can cause adverse drug reactions (ADR) or insufficient drug effects [2–4]. Approximately 20 genes affecting about 80 agents have been identified as actionable in the clinic [5], and the Clinical Pharmacogenetics Implementation Consortium (CPIC) updates continuously related guidelines with dosing instructions for relevant gene-drug combinations [6]. Nevertheless, the actual implementation of pharmacogenetic testing in clinical patient treatment is progressing slowly.

Relling and Evans summarized some general barriers for pharmacogenetic testing in clinical practice: lack of incentives to prevent adverse events; costs and complexity of computational approaches to identify and prioritize genetic variants that influence prescribing decision; disagreement among professional or guideline generating groups; and a lack of awareness of clinicians concerning pharmacogenetic effects. Further, they concluded that the costs associated with pharmacogenetics are shifting from the laboratory testing to the costs for generating evidence-based decisions from the genetic data [5].

Related to existing implementations, Hinderer et al. reviewed 20 pharmacogenetic decision support systems. Sixteen had prototype status and are often built for one

[1] Corresponding Author, Institute for Medical Informatics I4MI, Bern University of Applied Sciences, Quellgasse 21, 2500 Biel, Switzerland; E-mail: kevin.tippenhauer@bfh.ch.

institution. Only one of the productive systems (TreatGx) was integrated with the local EMR [7]. The same lack of EMR-integration is corroborated by the Ubiquitous Pharmacogenomics project (U-PGx), where only one out of the seven involved European countries implemented an active clinical decision support for pharmacogenetics [8]. Having not found any suitable and EMR-related solution, we developed a prototype implementation for assessing gene-drug-combinations in clinical oncology at the Department of Clinical Chemistry of the Bern University Hospital Switzerland (INSEL). Our goal was to derive a component-based approach to support the integration of pharmacogenetic clinical decision support into a broader range of clinical information systems (CIS).

2. Methods

Initially, we identified the main components required for our prototype: inference engine, knowledge base, and data dictionary (see also [9]). Next, we iteratively defined the architecture according to a typical workflow of pharmacogenetic analysis as described by clinicians at the INSEL. We chose the open source tool PharmCAT as the inference engine [10]. It generates pharmacogenetic reports in JSON and HTML from genetic data in the Variant Call Format (VCF) and comes with a set of CPIC guidelines (the clinical knowledge base) in a proprietary JSON format, which we adapted to our needs. To map information about described drugs we used the Swiss medication catalog "hospIndex" [11] which is incorporated in the INSEL EMR system. The link to the EMR user interface was implemented using the HAPI API, an open-source object-oriented HL7 2.x parser for Java [12]. The CDSS user interfaces were created with the proprietary parametrization language of the CIS within the medication module of the INSEL EMR system. In order to verify our prototype, a synthetic data generator for producing VCF data was implemented using the proprietary parametrization language of the INSEL EMR system.

3. Results

The underlying generic system architecture of our solution (Figure 1) considers two different workflows. The first workflow is triggered by the entry of new patient related genetic data and generates a report comprising the relevant pharmacogenetic findings (PGx Report Generator in Figure 1). It uses following components:

- The genetic data repository to store the raw genetic data.
- The knowledgebase with rules to annotate the genetic data.
- The data dictionary to encode the resulting report.
- The genetic annotation engine to generate the pharmacogenetic report.
- The pharmacogenetic report repository to store the report.

When a new genetic record is added to the Genetic Data Repository, the record is pushed to the Genetic Annotation Engine. This engine is supplied with the relevant pharmacogenetic rules from the knowledgebase and with the code mapping (e.g. translating phenotype "Poor Metabolizer" into a LOINC code) from the data dictionary (Knowledge Resources in Figure 1). Finally, the annotated report is added to the shared PGx report repository.

Once a PGx patient report is available (PGx Decision Support System in Figure 1), the second workflow can be triggered by a drug prescription event, which returns the case specific information from the corresponding CPIC guideline. It comprises the following components:

- The CIS, which sends the prescription and receives the decision support result.
- The pharmacogenetic report repository to read the patients report.
- The knowledgebase containing the decision support rules to evaluate the report for the given drugs.
- The data dictionary to encode the results.
- The CDSS-Engine to deduce the correct decision support message.

Upon a new drug prescription, the CIS sends a prescription event with the encoded prescription data to the CDSS-Engine. The CDSS-Engine looks up the encoded drug in the Data Dictionary and loads the corresponding guideline rules from the knowledgebase. If guideline rules are found, the CDSS-Engine tries to find an existing pharmacogenetic report of the patient. If none exists, a warning is sent to the CIS, since a potential pharmacogenetic adverse event cannot be excluded, and a genetic examination might be advisable. If a pharmacogenetic report of the patient exists, the CDSS-Engine applies the guideline rules on the report and returns applicable dosing instructions to the CIS. Finally, the CIS displays the received message to the prescribing physician.

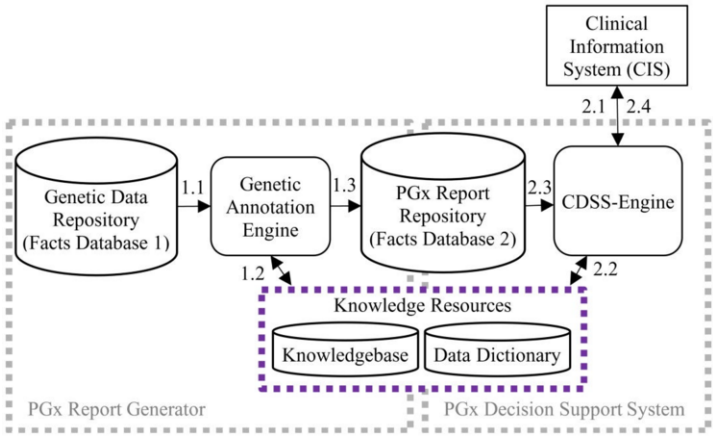

Figure 1. Generic model of a pharmacogenetic decision support system. A Genetic Data Repository acts as the fact database for the Genetic Annotation Engine and provides access (1.1) to the genetic data of the patients. The Genetic Annotation Engine creates a report with the relevant pharmacogenomic findings using the parameters from the shared Knowledge Resources (1.2). The report is stored (1.3) in a PGx Report Repository, which acts as the fact database for a second inference module (CDSS-Engine). Once the rules from the Knowledge Resources have been applied to the patients PGx report, the engine uses the drug prescription from the CIS (2.1) and returns a warning (2.4).

In our implementation, the PGx Report Generator uses a system folder with VCF files for the Genetic Data Repository. A modified version of PharmCAT acts as the Genetic Annotation Engine. The modifications enable PharmCAT to:

- Accept HL7v2 observation request messages as a trigger to read data from the Genetic Data Repository.
- Read the Data Dictionary in form of a configuration file containing the mapping between the genotypes/phenotypes/substances and their encoding.
- Encode genotypes, phenotypes and substances with codes from the Dictionary.

- Turn the pharmacogenetic report into a HL7v2 observation result.

The drug selection event in the CIS oncology drug prescription form triggers the PGx Decision Support System. The CDSS and the PGx Report Repository have been implemented directly within the CIS. The reports from the PGx Report Generator are stored in the database of the CIS acting as the PGx Report Repository for the CDSS-Engine. We implemented the knowledgebase and the data dictionary as internal CIS tables.

For generating synthetic data, we developed a Laboratory Data Simulator (Figure 2). It generates custom VCF data with user defined genetic variants and allows to validate our solution. The following validation workflow was realized in order to compare the issued message with the expected one:

- The validator selects variants to be tested within the laboratory data simulator. The selected variants are added to an existing VCF file template.
- The VCF file is added to the Genetic Data Repository and a HL7v2 observation request message triggers the PGx Report Generator to generate a report.
- The resulting report is returned as HL7v2 observation result to the Pharmacogenetic Report Repository of the CIS and the PGx CDSS is triggered.

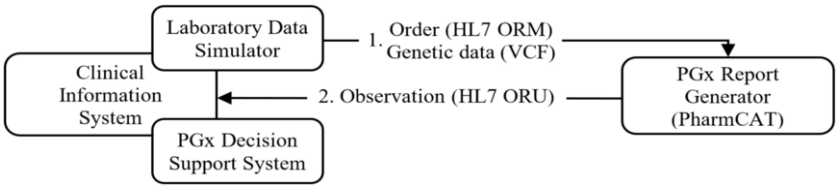

Figure 2. Our prototype integrated with the Laboratory Data Simulator (LDS). In the Clinical Information System (CIS), the user can invoke the LDS and generate a genetic data set for one of the test patients. When finished, the LDS triggers the PGx Report Generator with a HL7v2 ORM message (1). The PGx Report Generator reads the genetic data of the patient and creates a PGx report in the HL7 ORU format, which is then sent to the CIS (2). The user can now either switch to the medication prescription user interface and test the actual workflow integration or test the PGx Decision Support System output directly in the LDS. The system supports pre-test and post-test alerts.

4. Discussion

We presented a component-based architecture for genetic decision support during medication prescription. We could identify six generic components, namely a Genetic Data Repository, a Genetic Annotation Engine, a PGx Report Repository, a CDSS-Engine, a Knowledgebase for PGx rules and a Data Dictionary (see Figure 1).

Due to the following reasons, we consider an architecture with two different workflows (writing genetic reports and providing decision support) as mandatory: First, discussions with physicians revealed that access to the pharmacogenetic report is required to assess the content of the CDSS warnings. Second, the processing of large genetic data records would lead to increased response times without a report in which the drug related phenotypes are available. This makes our approach suitable for such large genetic data contexts.

The modularity of the system enables, a hospital to select components from the system architecture according to its requirements and plugging them together. Our prototype implementation does not completely match the proposed architecture. While the PGx Report Generator is indeed loosely coupled with the CIS using the HL7v2

standard, the PGx CDSS itself depends not only on the CIS in use, but also on its configuration within the hospital. In addition, we have not implemented shared knowledge resources. Instead, the PGx report generator and the PGx decision support system have each their own copy of the knowledge resources, which need to be updated simultaneously to ensure the correctness of the results. Such a solution is vulnerable to inconsistencies. To avoid this risk, the knowledge resources should be shared as depicted in Figure 1.

Using proprietary data formats, as we did to access the knowledge resources reduces the systems adoptability. However, the goal should be to provide a plug and play experience for the whole pharmacogenetic decision support system, as proposed, e.g. by the SMART Health IT movement [13]. Hence, a standardized data exchange model such as HL7 FHIR with the modules "Terminology" and "Clinical Reasoning" should be used for interfacing with the knowledge resources.

Acknowledgement: This work was supported by swissuniversities and Gebert Rüf Stiftung through the program «BREF – First Ventures».

References

[1] W. Bush, D. Crosslin, A. Owusu-Obeng, J. Wallace, B. Almoguera, M. Basford, et al., Genetic variation among 82 pharmacogenes: The PGRNseq data from the eMERGE network, *Clin Pharmacol Ther* **100** (2016), 160–169. doi:10.1002/cpt.350.

[2] L. Wang, H.L. McLeod, and R.M. Weinshilboum, Genomics and Drug Response, *N Engl J Med* **364** (2011), 1144–1153. doi:10.1056/NEJMra1010600.

[3] U.A. Meyer, U.M. Zanger, and M. Schwab, Omics and Drug Response, *Annual Review of Pharmacology and Toxicology* **53** (2013), 475–502. doi:10.1146/annurev-pharmtox-010510-100502.

[4] U.A. Meyer, Pharmacogenetics – five decades of therapeutic lessons from genetic diversity, *Nat Rev Genet* **5** (2004), 669–676. doi:10.1038/nrg1428.

[5] M.V. Relling & W.E. Evans, Pharmacogenomics in the clinic, *Nature* **526** (2015), 343–350. doi:10.1038/nature15817.

[6] CPIC Guidelines, (n.d.). https://cpicpgx.org/guidelines/.

[7] M. Hinderer, M. Boeker, S.A. Wagner, M. Lablans, S. Newe, J.L. Hülsemann et al., Integrating clinical decision support systems for pharmacogenomic testing into clinical routine - a scoping review of designs of user-system interactions in recent system development, *BMC Medical Informatics and Decision Making* **17** (2017), 81. doi:10.1186/s12911-017-0480-y.

[8] K. Blagec, R. Koopmann, M. Crommentuijn–van Rhenen, I. Holsappel, C.H. van der Wouden, L. Konta et al., Implementing pharmacogenomics decision support across seven European countries: The Ubiquitous Pharmacogenomics (U-PGx) project, *J Am Med Inform Assoc* **25** (2018), 893–898. doi:10.1093/jamia/ocy005.

[9] T. Bürkle, H-U. Prokosch & J. Dudeck, Steps towards Integration of Knowledge Based Functions into a Hospital Information System, *Studies in Health Technology and Informatics* **34** (1996), 286–290.

[10] pharmcat.org, PharmCAT. (n.d.). http://pharmcat.org/.

[11] HCI Solutions AG - hospINDEX, Daten für Spitäler, (n.d.). http://www.hcisolutions.ch/de/datenbanken-und-software/index-datenbanken/hospindex.php.

[12] HAPI – The Open Source HL7 API for Java, (n.d.). https://hapifhir.github.io/hapi-hl7v2/.

[13] SMART Health IT, (n.d.). https://smarthealthit.org/.