Representing Medication Guidelines for Use in Production Rule Systems in the Context of Polypharmacy

Jonas W. Könning*, Carlos A Velasco† and Yehya Mohamad†, Stefan Decker‡ and Oya Beyan‡
*RWTH Aachen University, Aachen, Germany
Email: jonas.koenning@rwth-aachen.de
†Fraunhofer Institute for Applied Information Technology FIT, Sankt Augustin, Germany
Email: {carlos.velasco},{yehya.mohamad}@fit.fraunhofer.de
‡RWTH Aachen University, Aachen, Germany
Email: {beyan},{decker}@dbis.rwth-aachen.de

***Abstract*—This paper presents an approach to represent medication guidelines in a machine readable form for its use within a home care environment for elderly patients with chronic diseases. The overall system comprises a patient-centred integrated care environment, supported by computerized systems to improve the quality of home hospitalization. One component of the system is a decision support system for improving the medication of elderly patients. For this purpose, a machine actionable version of standardized medication guidelines was required. However, such version was not available, the translation of the guidelines from human readable into machine readable rules and the implementation are complex and pose many challenges e.g. ambiguity of the rules. Scope of this work included the design and implementation of such a rule base. Guidelines were selected based on their prominence in the domain and analyzed for their structure to allow for the creation of templates, which can be used in the automatic generation of rules. The templates were designed to work with the Drools business rule management system. While still simple, the current prototype shows good performance and potential for future extensions.**

***Index Terms*—health information management, rule-based decision-making system, polypharmacy, drug delivery, drug-drug interactions, comorbidity**

I. Introduction

The problems originated by an aging population have deserved greater attention by a multitude of organizations, notably in the World Health Organization (WHO) report "Global Health and Aging" in 2011 [1]. The number of people aged 65 or older is projected to reach 1.5 billion in 2050, from 524 million reported in 2010. Together with the eradication of infectious diseases and parasites in most parts of the world, which were especially dangerous for infants and children, chronic noncommunicable diseases are now the biggest burden on health and on the health care systems. Elder patients tend to suffer from so called co-morbid illnesses (or *comorbidity*), i.e., the presence of additional diseases or disorders that exist concurrently with a primary disease. This is challenging in multiple aspects for health care professionals and leads to increased usage of health care resources [2][3]. This leads to more complicated and/or multiple concurrent treatments, which usually lead to long-term use of multiple drugs in combination [4][5], called polypharmacy. Countermeasures to issues arising from polypharmacy are urgently required, as polypharmacy is common in elderly patients [6] and home hospitalization requires systems to monitor possible complications to ensure patient safety and provide a better medication prescription. Therefore in this work, we focused on potential drug-drug interactions (Potentially Inappropriate Prescribing, PIPs), and developed a rule-based system to detect possible adverse events.

Polypharmacy is defined as the concurrent use of multiple (usually more than four) medications or, sometimes, as the unnecessary use of multiple and/or redundant medications [7]. As mentioned before, this is common in adults older than 65 years, which shows that generally more than half of all patients older than 65 years take more than 5 prescription drugs [6]. The situation is complicated further by over-the-counter medications. Studies regarding such medications show that, especially in certain communities, 90% of the patients take more than 1 and almost 50% take 2 to 4 of these freely available medications [6][8]. Additionally, because of incomplete case histories and cases of low patient compliance, the medical professionals treating the patient often have incomplete knowledge on which substances the patient is actually using. Patient safety is a problem area and topic of active research in general, as adverse drug events are a serious problem in modern health care. Multiple studies brought this to attention, notably the report "To Err is Human" in the US, however, adverse events are preventable in many cases [9][10][11].

Multiple clinical guidelines and screening tools have been developed to check for PIPs. Mark Beers et al. created a list of medications that can be considered inappropriate for older patients in long-term care in 1991 [12]. Beers' criteria were updated regularly and are the basis for other criteria sets, most notably "Screening Tool of Older Persons potentially inappropriate Prescriptions" (STOPP) and "Screening Tool to Alert doctors to the Right Treatment" (START). Both are evidence-based lists of criteria, first published in 2008 and developed in Ireland by a round of experts using the Delphi consensus method [13][14]. Version 2 of these criteria was published in 2014 [15]. STOPP/START resulted in much research interest, many countries and institutions support the tools and consider them appropriate for evaluating prescriptions [16]. Here is an example of the STOPP criteria: The following prescriptions

 ISBN: 978-1-61208-608-8

are potentially inappropriate to use in patients aged 65 years and older for cardiovascular system:

1) Digoxin for heart failure with normal systolic ventricular function (no clear evidence of benefit).
2) Verapamil or diltiazem with NYHA Class III or IV heart failure (may worsen heart failure).
3) Beta-blocker in combination with verapamil or diltiazem (risk of heart block).

And here is an example of the START criteria for the respiratory system:

1) Regular inhaled β 2 agonist or antimuscarinic bronchodilator (e.g. ipratropium, tiotropium) for mild to moderate asthma or COPD.
2) Regular inhaled corticosteroid for moderate-severe asthma or COPD, where FEV1 $<$50
3) Home continuous oxygen with documented chronic hypoxaemia (i.e. pO2 $<$8.0 kPa or 60 mmHg or SaO2 $<$89

However, none of these guidelines were available in a machine readable format, they are indented to be used manually by medical professionals, which can create a considerable workload. The usage of the guidelines out of paper documents is likely unrealistic due to time restrictions of the medical staff. There is an urgent need to translate such rules into machine readable form and integrate them into decision support system as part of the medication prescription process making them in almost real time available for medical doctors. So in the frame of this work a set of machine readable rules was created using the Drools rule engine [17]. The target of this work was to translate the STOPP/START guidelines into machine readable rules using the Drools format, wrapped them in a prototype service application. The developed prototype is being integrated with the home monitoring system allowing its extension beyond existing systems' capabilities and approaches especially when combined with real time sensors' data measuring the body vital signs and utilizing machine learning algorithms. production rule systems have been used together with other technologies in "business rule management systems" since the early 1990s, especially in industries with a lot of rules in everyday operation, such as insurances [18]. The RETE algorithm that optimizes the process of matching conditions to rules by "compiling" a network of conditions and their relation has been designed in the 1980s for such scenarios [19]. Derivatives and improvements of this algorithm are still used in current rule engines, such as Drools [20]. The usage of a decision support system to reduce medication errors shows good results, especially when used at the ordering stage of a medication [21]. A Business Rule Management System (BRMS) is software that creates, supports, and executes decision logic and business rules. Drools is one of the most used BRMSs being utilized by thousands of organizations currently. The object-oriented system is an augmented implementation of the known Rete algorithm tailored for the Java language. It includes both forward as well as backward chaining interference based rules engine and it provides a framework to allow business logic externalization in a common place. Efforts have been made to compute sets of medical guidelines for using them in applications. One example is STRIPA [22], a rule-based decision support system for medication reviews. It was developed with the Systematic Tool to Reduce Inappropriate Prescribing (STRIP) in mind, a drug optimization process, and aims at making the pharmacotherapeutic analysis step easier and less time-consuming by automation [22]. This system was not targeted to integration in home monitoring systems and was developed as a stand alone system.

In section 2, an elaboration on design principles, system design and involved frameworks is given. In section 3, the design implications, data model and the approach of implementation are described. In section 4, the evaluation methodology and the results are depicted. In section 5, conclusions and future work are discussed.

II. Design

This prototype was designed with usability focus, to have a structured and easily manageable representation of the rules, without losing too much precision in detecting rule violations or losing too much flexibility in the addition of rule conditions and the manipulation of rules. With this in mind, we looked at state-of-the-art rule engines with a wide implementation in the industry, such as Drools. Additionally, with the use of the Drools Rule Language (DRL), rules can also be generated from schematic representations, so called decision tables. This confirmed our choice of Drools as the core of the system, as it has also proven itself in similar applications [22].

A. System Design

As shown in Figure 1, the system was designed as a self-contained service, with two possibilities for interoperability with the rest of the home monitoring system in mind. One is the usage of the same database and framework as other related projects, Apache Cassandra and Apache Spark, so that the new service can be introduced into a new environment without changing much of the code and/or configuration. The second possibility is a REST API (via KIEServer) that can be used for sending data to the service for evaluation against the rule base and is designed to be easily extended. During development, the focus was set on the first possibility: the system was built and tested with the same tools and technologies that allow easy integration into home monitoring systems.

It is important to note that Drools is used in a stateless fashion. Stateless Drools sessions can be called like a function, a batch of data is passed to the session and the results of the rule execution are sent back. The production rule system does not keep track of (generated) knowledge and the result of one rule execution will never trigger or influence the execution of other rules. This was the desired operation mode in the use case at hand.

 ISBN: 978-1-61208-608-8

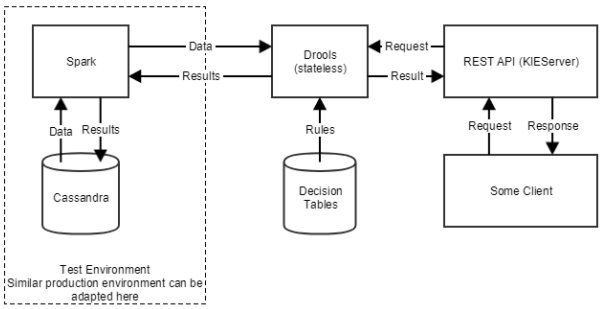

Fig. 1: Overview system design

B. Analysis of the Guideline Structure

In the early phase of the development, the STOPP and START sets of criteria were analyzed for their structure. As an intermediate result, the following parts could be identified in most STOPP rules:

- The rule subject, being a drug or drug family
- A part specifying co-medication that might interact with the subject
- Therapeutic information / information on pharmacotherapy, filtering for special (mis-)use cases of the subject
- A diagnosis / treatment condition, in some cases the subject is only harmful / not harmful if some condition, symptom or treatment is present
- Some ”dependent clinical characteristic”, an additional condition that can be a diagnosis, symptom or lab value for example that narrows down the execution of the rule (often exceptions to the rest of the rule)
- The outcome: in the case of STOPP rules that is a warning to consider a medication change for the patient

START rules have similar parts that can be identified, but they do not contain a subject in the sense above, meaning a drug or drug group that has to be matched with a drug from the patient’s records for the rule to be executed. Instead, they contain a drug that they advice in case the rule is executed successfully, as START aims to recommend initial and/or additional medication that is proven to be beneficial in the case described in the rule.

The various connective words and phrases used in the plain English statements of the original set of criteria could be reduced to ”and”, ”or” and ”not”. Statements containing temporal modifiers like ”X with concurrent Y” were also reduced to the logical operators above. However, at present this first prototype does not have the required structured data from electronic health records to execute that specific type of rules.

The identified rule parts were also further sub-divided into ”types”. Conditions on drug dosage, time of prescription and co-therapy could be reduced to conditions on intolerance, efficacy, duration, dosage and a check for contraindication. Further time/co-therapy conditions were: ”The subject is the first treatment for something”, ”used as a long-term treatment”, ”used as secondary prevention”, ”used when an alternative is available”, ”used as mono-therapy”, ”used instead of some other drug” and ”used with some other drug of the same class”. Similarly, detailed types of symptom / diagnosis conditions were: ”Usage of the subject drug with a history of some diagnosis X”, ”used as a treatment for X”, ”used for therapy in X” and ”used unless concurrent X”. Clinical characteristics could be reduced to: Health conditions, physical examination result, interventions, disease history, laboratory results.

III. Design Implications and Implementation

In principle, it would have been possible to create conditions in the machine readable version of the criteria sets for all of these condition types. However, as such a level of detail was not available in the data and is difficult to achieve and use correctly, many types were left out or combined into very basic conditions. Some information is contained in the codes of the classification systems used and a good selection of codes for condition checks allows representation of some of these detailed condition types in their more general parent condition. With more detailed data, additional conditions can be implemented, but some only appear in very few and quite specific rules, it might be better to accept the possibility of false alerts and let the medical professional decide if acting is necessary, instead of trying to make a rule more precise with complicated conditions on unreliable data. This trade-off had to be evaluated throughout the design and development process. It will also be important for future improvements and extensions. For the realization of the concept, proven technologies have been chosen. The requirements fit the use case of a production rule system. Additionally, a general purpose programming language, such as Java was chosen as a consequence of the choice of the rule engine.

A. Data Model

As Drools is data-driven, the data model is very important. It consists of a typical object-oriented programming class hierarchy: all used objects are plain Java objects. Each type that was used in one of the decision tables is represented by a Java class, which includes objects representing patients, drugs and diagnosis. As shown in Figure 2, the system is currently only using coding systems for both diagnosis and drugs, the presented objects basically only act as a container for these codes with some additional functionality.

The defined objects contain methods for matching codes and code prefixes by simple string matching. This way, the rules can easily take advantage of the structure of the mentioned coding system. For example, if a guideline from the STOPP set states that all opioids should never be given together with some other drug, we can take the common prefix for opioids in the Anatomical Therapeutic Chemical (ATC) system and use it for pattern matching in the rule, by just passing it system for evaluation. This is easier than compiling lists of drugs manually and less error prone, but not as simple as it may sound, in fact, it is quite difficult in many cases to find a

 ISBN: 978-1-61208-608-8

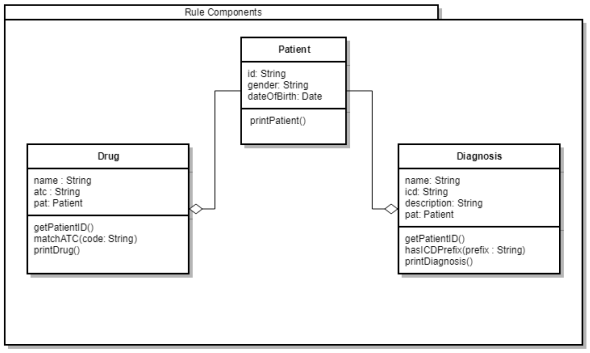

Fig. 2: Overview system design

coded representation of what is stated in the original criteria that means exactly the same semantically. Still, we will see how rules make use of the coding systems in detail in the next sections.

B. Rule Structure and Decision Tables

Each rule can be fitted to a schema consisting of conditions. Basically, as mentioned above, the following parts could be identified as a general structure in all STOPP/START criteria: a rule subject, being a drug or drug set, another drug or drug set, representing drugs that might interact with the subject (co-medication group), a set of diagnosis and a patient, to which all other objects have to relate.

The co-medication group filters for drug-drug interaction using ATC codes, many STOPP rules state something like: ”If the patient is taking drug A and he is also taking drug B at the same time, revoke the prescription for drug B”. The assumption was made that drugs supplied to the system are always prescribed at the same time, there is currently no check for concurrency. This is due to the fact that the data that was available at the time of writing did not include such details. We have examples of how the rule base could be made more precise if the available data is of better quality, something that will become apparent multiple times and has already been mentioned before.

The diagnosis group filters for diagnoses, symptoms, or other information that can be represented by the International Classification of Diseases (ICD) system. This further narrows down the execution of the rule immensely by applying an additional condition that many of the rules have in common. To summarize, the most important criteria and parts of the resulting decision tables are the rule subject, drug interactions, and coded diagnoses, they were the only ones that were selected for implementation.

The observation that most of the rules in both STOPP and START follow a certain scheme lead to the belief that they can be reduced to a fixed structure, basically a prototype rule with parametric conditions. That is why so called decision tables were chosen as the source of rules for the system. Drools supports Microsoft Excel spreadsheets with a certain structure as an input and will generate rules from them. This approach has certain advantages over representing the rules in files using the decision support language described above in certain use cases and the STOPP/START criteria were quite compatible with the approach.

Figure 3 represents a first iteration of a STOPP table. Each rule follows the same structure, as it is given by the decision table. First, the knowledge base is checked for the drug that is the subject of the corresponding STOPP guideline. If the subject drug is found, the following columns contain all other tests, but not always all tests for every rule, some are left out, if the guideline does not contain such a requirement. This can be done by leaving the respective cell empty. The conditions check for interacting drugs, for diagnosis and that the patient ID of both of these objects match the one of the subject drug. We do not verify the patient’s age, although the STOPP/START criteria are designed for people above the age of 65, because the rule base will be used in an environment, where only data of such patients will be processed. An additional column contains the negation of the check for interaction drugs. This was necessary for a few rules, which apply only if a certain drug or drug class is not in the medication plan of a patient. This, of course, leads to the logical conjunction turning into a logical disjunction: the condition requires the medication list of a patient to be free of all the mentioned codes or code prefixes. Note that this generally is a first naive table layout and does not take performance optimization into account. The order of the conditions can be optimized for quicker execution, as we will see, but this always affects all rules, one drawback of using decision tables. If the rule is evaluated to be positive, the marked action of the available ones is taken. In Figure 3, this is one of four very basic output variants. During use of the system later on, this can be any kind if post-processing or event handling.

Decision tables make creating, testing and updating of a larger rule base of similar rules easier. Once the structure of the decision table and the template code is done, only parameters have to be entered into the table. In the case of the rule base presented here, this brings other challenges, mainly in choosing the correct codes for representing the symptoms and illnesses mentioned in the STOPP/START guidelines, but the actual implementation work is reduced. In the case of the ”subject drug” test template code, the rule engine will loop over all parameters found in the rules cell for this condition and that the results of the evaluation of each single one should be connected with the ”or” operator. The parameter is inserted at the placeholder ”$”, in this case we pass it to a function. The template code section allows us to call functions of the object specified in the row for types above. In this case, the function ”matchATC” is called with the parameter inserted in the rule’s cell. We can also see that we can add arrays of parameters to rules by just adding them into the cell, separated with commas, in this case strings representing the ATC code or ATC code prefix we want to match. The method we call is described in the data model section, which we use it to match codes

 ISBN: 978-1-61208-608-8

RuleTable STOPP

NAME	CONDITION	CONDITION	CONDITI	CONDITION	CONDITION	ACTION
	$s : Drug	$drug : Drug	$drug : Dr	$p : Patient	$d : Diagnosis	
	forall(||){matchATC($)}	getPatientID() == $s.getPatientID(),forall(||){matchATC($)}	forall(&&){matchATC($)}	(id == $s.getPatientID())	getPatientID() == $p.id, forall(||){$}	System.out.println("------ STOPP WARNING ------"); System.out.println("$param"); $s.printDrug(); $p.printPatient(); $d.printDiagnosis(); System.out.println("------ END STOPP ------");

NAME	subject drugs	complication drugs	neg. complication drugs	patient	diagnosis	OutputNoDrug
STOPP B1	"C01AA05"			TRUE	hasICDPrefix("I50")	STOPP B1: No clear evidence of benefit
STOPP B2	"C08DA01","C08DB01"			TRUE	hasICDPrefix("I50.04"),hasICDPrefix("I50.05"),hasICDPrefix("I50.13"),hasICDPrefix("I50.14")	STOPP B2: May worsen heart failure

Fig. 3: The template code of the STOPP decision table

and code prefixes. The template code for the ”complication drugs” condition is the same, but adds an equality check for the patient ID to make sure that only drugs that are actually taken by the patient are considered. The template code for the diagnosis check, basically works in the same way. But instead of inserting a parameter into a method call, we just specify that all parameters should be connected in a disjunctive fashion. The rules' fields of this condition check contain both method calls and equality checks. The same result is achieved in a different fashion. This basically concludes the description of the pre-optimization STOPP table.

The decision table for the START criteria works in a similar way as the templates, but has different requirements for the number and order of condition checks and for the patient check. This is because we do not have a subject drug in every case that can be used to get all patients taking the drug like in the STOPP decision table. Instead, we now have to look at all patients. In the next step, all drugs with a matching patient ID are checked against the rule's codes and code prefixes, just as in the STOPP table. The diagnosis check is also the same, the number of parameters is just higher in many cases.

C. Integration into the home monitoring system's workflow

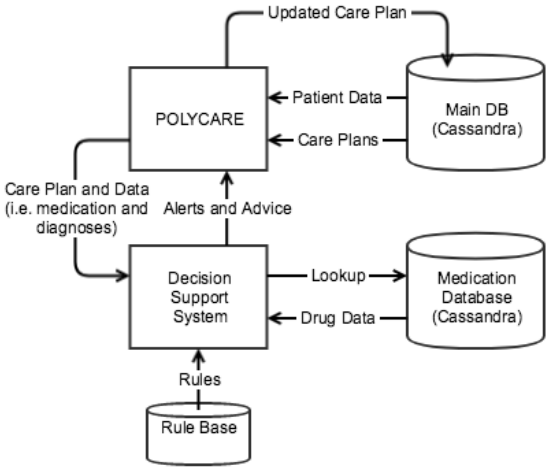

Fig. 4: Overview of integration of the prototype in the home monitoring system Polycare

The system will take in information about a patient and provide feedback about drug interactions and interference while under certain therapies or suffering from certain diseases, as defined in the STOPP/START criteria. Depending on whether a START or STOPP condition is detected, an alert is given or a recommendation for therapy is given. The information is sent back to the subsystem developed for persistence in a database and to be used for care plan creation, as seen in Figure 4. Not shown in the workflow diagram is the process of looking up medications in the Apache Cassandra database. A ”Pharmazentralnummer” (PZN, meaning ”central pharmaceutical number, a standardized number to identify medications and medical products in Germany) id, from the electronic health record of a patient is used to get detailed information about a drug (more specifically about the active substance) and its classification in other coding systems such as ATC, which is currently used in the system so far.

Finally, information about the medication of the patient and her diagnosis is used. If available, more precise information such as dosage of specific drugs, duration of the treatment and lab values can be used to give better feedback. But in this use case, it is better for the patient if a false alert is raised than if no alert is raised due to incomplete knowledge.

IV. Evaluation Methodology and Results

There are technical measurements and results, such as performance and a possible performance differences after optimizing the initial system and rule base. Additionally, there is a discussion on how the results presented here fit in within the project infrastructure, what drawbacks exist and what may need more work. An issue is the selection of both ATC and especially ICD codes for the rule's conditions. A first selection was used in this work to demonstrate how such a rule base for the STOPP/START criteria and the accompanying system might work, but there was no guarantee for correctness of the rules from a medical standpoint at any time.

Technical evaluation resulted in some key points. Most obviously, as previously mentioned, finding the correct order of condition checks and was part of the task at hand, but turned out to be marginally important for performance of the system. Still, with more conditions and more complex comparisons one should think about the structure of the table again, as it

 ISBN: 978-1-61208-608-8

is important to narrow down the set of possible matches for a rule by applying the simplest conditions that exclude the most facts first. In the case of the START table, there was some performance gain by filtering the diagnosis codes first, as most START rules do not have a drug condition and filtering by diagnosis can exclude a case early on. This also applies to choosing the right starting condition for the rule. In the case of the START table, we unfortunately have to look at the patients in all cases to see if there is a beneficial prescription for them. But the STOPP table can match a drug a patient is taking in the first step. If the data set sent to the knowledge base does not contain a matching code, it can be discarded immediately.

It is important to note that the knowledge base had to be created to work in a stateless fashion: the working memory does not keep track of facts other than initial ones created at the time of starting the system. Incoming data is processed, rules are matched and executed and some action is taken. Afterwards, the data is discarded and the knowledge base remains unchanged. This also means that we can only look at one (the system could be configured to cache data and process batches) set of data representing a patient's record at the time. If the knowledge base were to be created to work with stateful execution, optimizing the rule base and the accompanying system would be a completely different task.

The use of stateless sessions also has some implications for the correct usage of the system. All data belonging to one patient should be inserted and executed in the same batch or at least all the patient's data from the same time frame. Otherwise, it would be possible, for example, to miss a drug-drug interaction as the system does not keep track of knowledge inserted or created during runtime. As an example, if the patient is taking a beta-blocker drug, we know this because of some report and an entry for this was created in a database, but the patient is also taking Verapamil and a separate entry was created for this, it might be possible that we miss the interaction between the two drugs, if there is no pre-processing step that makes sure that all entries for one patient and in the same time frame are collected before sending the data to the rule engine. This has to be kept in mind to get good results from the system.

As mentioned briefly beforehand, the system's performance in terms of accuracy, is determined by the choice of codes to represent the conditions of the original guideline statement. This is especially true, as the technical correctness of the rules was verified using unit testing, which covered nearly all of the cases which might trigger a rule evaluation. But the correctness of the result in a medical sense, of course, still depends on the codes used for the condition checks. The diagnosis codes used in this were chosen by looking at the ICD-10-CM index and choosing codes that seemed suitable. Choosing ATC codes was easier, as it is pretty clear in most cases which substances are mentioned in the original guideline statements. But as the ATC structure allows one substance to have multiple codes if it can be used for different treatment goals on different physiological systems, one has to choose the correct code or include all variants. Generally, it is always possible to include more codes, especially in the case of diagnosis codes, to catch more rule violations. But this can lead to more false alarms. Working around the coding systems by, e.g., compiling a list of single codes instead of using the hierarchy of the system, is generally more difficult and inefficient than using well-defined codes or the structure of the coding system, but sometimes there is no other way.

Some rules from the STOPP/START sets were especially difficult to represent by an entry in the respective decision table. Rule STOPP D10 for example, warns that neuroleptics should not be used as hypnotics. As there is no ATC group that contains all neuroleptics, a list of codes had to be created manually. Modafinil, sodium oxybate and methylphenidate were used as the codes for the subject drug check. But this must be confirmed and corrected by medical professionals yet. More difficult and quite representative of the issues in representing the criteria as formal rules is the condition that these neuroleptic medications must be stopped if they are used as hypnotics. There is no information about the treatment goals or the intent of the medication available in the test data sets, and this information will also not be available for the prototype development. To catch all cases that might put the patient in danger, a compromise had to be made: within our first implementation, all neuroleptics will trigger rule STOPP D10, even if their usage is correct and not dangerous to the patient. The physician that receives warnings from the system will have to decide, whether the warning is correct or just a false alarm. Similar decisions had to be made for most rules that had some kind of exception or modifier in their original statement.

Most recommendation systems and similar applications have to find a balance regarding this issue. In our case, if the number of false alarms is too high, the actual value of having an automatic system for evaluation of the STOPP/START guidelines is reduced, as the amount of work compared to manually checking the criteria is not small enough to justify using the system. Similarly, if the number of cases missed by the system that actually put the patient in danger is too large, there is also no benefit in using it.

The rule base and system presented here will be evaluated and refined with these criteria in mind in the future, after integration together with the rest of the home monitoring infrastructure. The used codes will need to be checked again. There already have been panels [23] that worked on finding codes for STOPP/START and they can be used together with new consensus finding methods and a group of medical experts to agree on a set of codes for the rules. A study on the acceptance of the whole system by medical professionals and patients can further show if the selection of codes and the knowledge base in general is too strict or too inaccurate. Feedback and experience with the system in a productive environment might even be more efficient than the theoretical selection of codes and conditions.

 ISBN: 978-1-61208-608-8

V. Conclusion and Future Work

There are multiple things to take away here and that left an impression on us during the development of the rule base. First, there is a big selection of technologies and software freely available for such a task, with active development and very prominent credentials. Their features were not really used to the fullest extend, but as the choice matches what is used in other parts of home monitoring systems, one can say that scaling the project will not be a problem. Drools also seems to be a good choice, it can create or read rules from multiple sources with flexible and powerful template coding and accompanying tools that can be used to modify rules even by people who do not know a lot about programming. It allows the use of the knowledge base as a server, but can also be integrated into any other form of Java application.

Using decision tables to represent the STOPP/START criteria was the correct choice. The criteria turned out to be surprisingly similar in their structure, allowing the use of template code to generate rules. Drools is also quite efficient in matching and executing rules. Running both the 77 rule STOPP and 34 rule START set against 10.000 test records took less than 1 second per case on a consumer grade laptop. However, in the case at hand, changing the structure of a table, for example the order of the rule's condition, did not improve performance significantly. In production use, it would be advisable to focus on the efficient usage of Apache Spark and Apache Cassandra, as the code for database access, data manipulation and analysis will impact performance the most.

While the decision tables are not too complex, the complexity of the problem lies in choosing the correct codes for both drugs and diagnosis. This is not a trivial problem and it impacts the whole domain. As many other authors have stated in their papers on similar projects and on electronic health records, standards for medical data, improved standards for coding and agreements on translating between them and from natural language would help the whole domain immensely. The rules that were created during the development of this work were not evaluated for their "medical correctness". We have seen that choosing codes that precisely represent what was stated in the STOPP/START criteria is difficult. The original statements are sometimes subject to interpretation itself. For any future development, it would be advisable to have a consensus process with medical experts to agree on coding, as many other projects did. Until there is an "official" coded version of the original criteria, there is no other possibility if coded data is to be used. One of the main conclusions drawn from studies [24] is that the quality of the available data is one of the biggest factors in the success of using such guidelines with a decision support system.

With these limitations and problems in mind, the presented rule base can still be a starting point for the decision support system. It will certainly be improved and evaluated further in the next development rounds. Many extensions are conceivable, for example the use of the Resource Description Framework (RDF) for representing rules built from conditions was one idea that came up during development. Giving feedback to the used standards and contributing to them is also important, in this case a Fast Healthcare Interoperable Resources (FHIR) resource could be devised for medical alerts. Another extension could consist of additional conditions to make rules more precise. Lab values, for example, are mentioned in certain criteria of the STOPP/START criteria and could be compared to lab values from a data source. But again, semantic equality has to be ensured, a variety of abbreviations and codes for lab values have to be translated to match the data source. Instead of relying on coded data, one could also make use of a natural language processing system that matches certain terms from written reports of the patient and can also infer the context in which some information is stated. Natural language processing systems are not new to the domain and have shown promising results, surpassing the use of coding in sensitivity [25]. Such systems can be a viable alternative to the use of administrative data and codes.

Finally, it is evident that the quality of underlying data and the systems used to structure it determine what can be done on top of them. The best decision support system in the world can only be as good as the input data it works on, even if methods to further infer knowledge are used. In general, as the quality of health care data increases and as more processes, data formats, workflows and other technologies are standardized, it will become easier to build systems giving warning and advice regarding medication that work with data from electronic health records.

Acknowledgment

This work was partially funded by the European Commission H2020 Programme (Grant Agreement 690367, POLYCARE: Poly-stakeholders integrated care for chronic patients in acute phases). The authors would like to acknowledge the support of the POLYCARE consortium for this work.

References

[1] World Health Organization, "Global health and aging," Geneva: World Health Organization, 2011.

[2] B. Starfield, K. W. Lemke, R. Herbert, W. D. Pavlovich, and G. Anderson, "Comorbidity and the use of primary care and specialist care in the elderly," The Annals of Family Medicine, vol. 3, no. 3, 2005, pp. 215–222.

[3] T. Lehnert et al., "Health care utilization and costs of elderly persons with multiple chronic conditions," Medical Care Research and Review, vol. 68, no. 4, 2011, pp. 387–420.

[4] K. Moore, W. Boscardin, M. Steinman, and J. B. Schwartz, "Patterns of chronic co-morbid medical conditions in older residents of us nursing homes: differences between the sexes and across the agespan," The journal of nutrition, health & aging, vol. 18, no. 4, 2014, pp. 429–436.

[5] G. Kojima, "Prevalence of frailty in nursing homes: A systematic review and meta-analysis," Journal of the American Medical Directors Association, vol. 16, no. 11, 2015, pp. 940–945.

[6] E. R. Hajjar, A. C. Cafiero, and J. T. Hanlon, "Polypharmacy in elderly patients," The American journal of geriatric pharmacotherapy, vol. 5, no. 4, 2007, pp. 345–351.

[7] S. P. Stawicki and A. Gerlach, "Polypharmacy and medication errors: Stop, listen, look, and analyze," Opus, vol. 12, 2009, pp. 6–10.

 ISBN: 978-1-61208-608-8

[8] S. I. Haider, K. Johnell, G. R. Weitoft, M. Thorslund, and J. Fastbom, "The influence of educational level on polypharmacy and inappropriate drug use: A register-based study of more than 600,000 older people," Journal of the American Geriatrics Society, vol. 57, no. 1, 2009, pp. 62–69.

[9] E. C. Davies et al., "Adverse drug reactions in hospital in-patients: a prospective analysis of 3695 patient-episodes," PLoS one, vol. 4, no. 2, 2009, p. e4439.

[10] C. Kongkaew, P. R. Noyce, and D. M. Ashcroft, "Hospital admissions associated with adverse drug reactions: a systematic review of prospective observational studies," Annals of Pharmacotherapy, vol. 42, no. 7-8, 2008, pp. 1017–1025.

[11] J. Lazarou, B. H. Pomeranz, and P. N. Corey, "Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies," Jama, vol. 279, no. 15, 1998, pp. 1200–1205.

[12] M. H. Beers et al., "Explicit criteria for determining inappropriate medication use in nursing home residents," Archives of internal medicine, vol. 151, no. 9, 1991, pp. 1825–1832.

[13] P. Barry, P. Gallagher, C. Ryan, and D. O'Mahony, "START (screening tool to alert doctors to the right treatment)—an evidence-based screening tool to detect prescribing omissions in elderly patients," Age and ageing, vol. 36, no. 6, 2007, pp. 632–638.

[14] P. Gallagher, C. Ryan, S. Byrne, J. Kennedy, and D. O'Mahony, "STOPP (screening tool of older person's prescriptions) and START (screening tool to alert doctors to right treatment). consensus validation." Int J Clin Pharmacol Ther, vol. 46, no. 2, Feb 2008, pp. 72–83.

[15] D. O'Mahony et al., "STOPP/START criteria for potentially inappropriate prescribing in older people: version 2," Age and Ageing, vol. 44, no. 2, 2015, p. 213. [Online]. Available: http://dx.doi.org/10.1093/ageing/afu145

[16] B. Hill-Taylor et al., "Application of the stopp/start criteria: a systematic review of the prevalence of potentially inappropriate prescribing in older adults, and evidence of clinical, humanistic and economic impact," Journal of Clinical Pharmacy and Therapeutics, vol. 38, no. 5, 2013, pp. 360–372. [Online]. Available: http://dx.doi.org/10.1111/jcpt.12059

[17] D. Project, "Drools documentation," 2017, [retrieved: January, 2018]. [Online]. Available: http://docs.jboss.org/drools/release/latest/drools-docs/html_single/#d0e32

[18] S. Hildreth, "Rounding up business rules," Computerworld, vol. 39, no. 21, 2005, p. 25.

[19] C. L. Forgy, "Rete: A fast algorithm for the many pattern/many object pattern match problem," Artificial Intelligence, vol. 19, no. 1, 1982, pp. 17 – 37. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0004370282900200

[20] M. Proctor, "R.I.P. RETE time to get PHREAKY," 2017, [retrieved: January, 2018]. [Online]. Available: http://blog.athico.com/2013/11/rip-rete-time-to-get-phreaky.html

[21] R. Kaushal, K. G. Shojania, and D. W. Bates, "Effects of computerized physician order entry and clinical decision support systems on medication safety: a systematic review," Archives of internal medicine, vol. 163, no. 12, 2003, pp. 1409–1416.

[22] M. C. Meulendijk et al., "Computerized decision support improves medication review effectiveness: An experiment evaluating the STRIP assistant's usability," Drugs & Aging, vol. 32, no. 6, 2015, pp. 495–503. [Online]. Available: http://dx.doi.org/10.1007/s40266-015-0270-0

[23] D. A. de Groot et al., "Specifying icd9, icpc and atc codes for the STOPP/START criteria: a multidisciplinary consensus panel," Age and Ageing, vol. 43, no. 6, 2014, pp. 773–778. [Online]. Available: http://dx.doi.org/10.1093/ageing/afu075

[24] M. Price, I. Davies, R. Rusk, M. Lesperance, and J. Weber, "Applying STOPP Guidelines in Primary Care Through Electronic Medical Record Decision Support: Randomized Control Trial Highlighting the Importance of Data Quality," JMIR Med Inform, vol. 5, no. 2, Jun 2017, p. e15.

[25] H. J. Murff et al., "Automated identification of postoperative complications within an electronic medical record using natural language processing," JAMA, vol. 306, no. 8, 2011, pp. 848–855. [Online]. Available: http://dx.doi.org/10.1001/jama.2011.1204

 ISBN: 978-1-61208-608-8