## A RULE-ENGINE-BASED APPLICATION FOR OVER-THE-COUNTER MEDICATION SAFETY

by

#### SARABPREET KAUR DHILLON

(Under the Direction of Lakshmish Ramaswamy)

#### ABSTRACT

Fatal Accidents involving prescription and over the counter medications have risen at a startling rate over the last 15 years. In the United States, statistics from the Centers for Disease Control and Prevention (CDC) indicate that there were 38,329 fatal drug overdoses in the year 2010, more than double the number observed in 1999. In this thesis, we explore a rule-engine based approach for personalized over-the-counter (OTC) medication safety advisory application. Our application is one of the first attempts towards providing timely advice to a patient on whether it is safe to ingest a particular medication at a given time. This application, which is driven by a rule-engine, takes into account multiple factors such as patient's health conditions, demographical characteristics and recent drug consumption histories. We present several experiments for studying the effectiveness and efficiency of our system.

INDEX WORDS: Healthcare, Rule Engine, patient safety, over-the-counter drugs

# A RULE-ENGINE-BASED APPLICATION FOR OVER-THE-COUNTER MEDICATION SAFETY

by

#### SARABPREET KAUR DHILLON

B.Tech, Punjab Technical University, India, 2007

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial

Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

© 2014

Sarabpreet Kaur Dhillon

All Rights Reserved

# A RULE-ENGINE-BASED APPLICATION FOR OVER-THE-COUNTER

### MEDICATION SAFETY

by

## SARABPREET KAUR DHILLON

Major Professor: Committee: Lakshmish Ramaswamy John A. Miller Tianming Liu

Electronic Version Approved:

Julie Coffield Interim Dean of the Graduate School The University of Georgia December 2014

## DEDICATION

I would like to dedicate this work to my family for their unconditional love and support.

#### ACKNOWLEDGEMENTS

I would like to express deep gratitude to my major advisor Dr. Lakshmish Ramaswamy who gave me an opportunity to be part of this project that will have a positive impact on the society. His guidance and motivation has been vital throughout my research and also, in writing my Thesis.

I would like to extend my warm thanks to Dr. John A. Miller and Dr. Tianming Liu for being part of my Thesis committee and providing their support and guidance throughout the M.S program.

Special thanks goes to my team member Michael Scott for his help, assistance and support throughout this project.

### TABLE OF CONTENTS

Page	
ACKNOWLEDGEMENTSv	ACKNOW
LIST OF TABLES	LIST OF 7
LIST OF FIGURESix	LIST OF I
CHAPTER	CHAPTE
1 INTRODUCTION1	1
Drug Accident Causes1	
2 PRIOR EFFORTS AND MOTIVATION4	2
Role of Federal regulatory agencies4	
Mobile health (mHealth) apps5	
Drug Interaction Checkers7	
Clinical Decision Support Systems7	
3 EXPERT SYSTEMS9	3
Architecture of a rule-based system9	
Jess Rule Engine11	
4 DESIGN OVERVIEW13	4
Design Features13	
Functional Overview14	
Drug Safety Criteria15	
Safety Rules Design15	
5 SYSTEM ARCHITECTURE	5

	Active ingredient retrieval module	23
	Multi-criteria drug-safety rule engine	24
	Advice Convertor	24
	The User Interface	25
6	EVALUATION	28
	Safety warning in textual form vs. Java code vs. Jess Rules	28
	Rule Engine Response time	31
	End-to-end Response time	
7	CONCLUSION	35
	Significance and broader impact	35
	Future Work	35
REFEREN	NCES	37
APPEND	ICES	
А	OTC DRUGS USED IN THE APPLICATION	47
В	SAFETY RULES FOR OTC DRUG - DOLOGEN	48

## LIST OF TABLES

Page

Table 1: Rule Engine Response Time (Experiment I)	32
Table 2: Rule Engine Response Time (Experiment I vs. Experiment II)	33
Table 3: End-to-End Response Time	34

## LIST OF FIGURES

	Page
Figure 1: The architecture of a rule-based system	10
Figure 2: The System Architecture Diagram	22
Figure 3: Register Screen	25
Figure 4: Login Screen	25
Figure 5: User profile Screen	26
Figure 6: Edit Account	26
Figure 7: Safety Checker	26
Figure 8: CAUTION message	26
Figure 9: No safety concerns	27
Figure 10: UNSAFE message	27
Figure 11: Rete Network for Dologen safety rules	

#### CHAPTER 1

#### INTRODUCTION

Fatal Accidents involving prescription and over-the-counter (OTC) medications have risen at a startling rate over the past 15 years. In the United States, statistics from the Centers for Disease Control and Prevention (CDC) indicate that there were 38,329 fatal drug overdoses in the year 2010, making it double the number observed in 1999 [1, 2]. About 48,000 women have died from painkiller overdoses between the year 1999 and 2010, an increase of 400 percent over the period [3]. Thomas Frieden, the head of the Centers for Disease Control and Prevention (CDC) remarked in February of 2013 "This is a big problem that has gotten worse quickly" [4]. Drug related incidents are now the leading cause of accidental death in the United States, accounting for more deaths than traffic fatalities or suicides, and causing more than 120,000 emergency visits in 2010 [65, 66]. This imposes significant overheads on the health-care system (both financial and human resources).

#### 1.1 Drug Accident Causes

Several reasons account to this overwhelming increase of drug related accidents. *First*, the patients may ignore the warnings on the labels because of their long textual form (overwarnings) or they might be unable to understand them properly. The warnings in the package inserts being non-patient-specific, i.e., un-personalized, patients may have difficulties in finding the information that is relevant to them. It has been found that around 60% of people cannot identify the active ingredient in their medication [5].

Secondly, lack of awareness amongst the public about prescription and OTC drug abuse and overdose is another contributing factor. About 40% of Americans believe that OTC drugs are too weak to cause any real harm [5]. *Thirdly*, patients may forget the recently taken medications or their ingredients, or they may be unable to keep track of the times they took their last medicine. The *fourth* factor that contributes to pharmaceutical accidents is that certain active ingredients are present in larger numbers of marketed drugs. Taking multiple medications that have common active ingredients might lead to an unintended overdose, e.g., Acetaminophen is present in more than 600 marketed drugs [6]. The *fifth* factor is concomitant administration of drugs or foods causing drug-drug interactions or drug-food/beverage interactions. Some drugs may render other drugs less effective when used concomitantly for example interaction between low dose aspirin (81 mg per day) and ibuprofen, making aspirin less effective when used for its anti-platelet cardio protective effect [7, 8, 9]. Sixth, high risk of drug interactions in older population because they have multiple chronic disorders such as high blood pressure, diabetes, and arthritis that require treatment and so are generally more likely to experience side effects. It has been found that older adults take four or five prescription drugs and two nonprescription (OTC) drugs each day on average, and many OTC drugs are potentially unsafe for them [11, 12, 13]. For example, most antihistamines, such as diphenhydramine, are "sedating" antihistamines and are found in many pain relief, cough and cold medications. These antihistamines may cause drowsiness or fatigue, leading to falls and injury, also worsening some of the disorders common among older people, such as glaucoma, etc [10]. Seventh, there is lack of effective communication channels by which FDA and other regulatory agencies can update the public about the

new drug warnings or safety alerts. Existing channels like TV commercials, websites may not be enough to attract sufficient attention of the public towards the essentially important drug safety information [14, 15].

As self-care with over-the-counter (OTC) medicines has evolved, the switching of prescription to nonprescription (OTC) medications has led to some important benefits like creating more self-reliant consumers, expanding markets for pharmaceutical companies but at the same time has become a source of potential health threat in the form of increased count of pharmaceutical accidents. This is because the safety determination for OTC drugs is solely the responsibility of the patients or their caregivers. This is unlike prescription medicines, where it is reasonable to assume that a doctor would have determined the drug to be safe for the patient in the context of patient's overall health situation, and the doctor or a pharmacist would have clearly explained how and when to take the drug.

#### CHAPTER 2

#### PRIOR EFFORTS AND MOTIVATION

Several efforts have been made towards mitigating pharmaceuticals accidents. The federal government has always been taking regulatory steps and developing services to ensure public safety in the face of rising drug related accidents. Moreover, informatics can play a vital role in controlling the alarming rise in pharmaceuticals accidents. A study by Epocrates indicates that online drug information systems help prevent more than 27 million drug interactions each year [16].

2.1 Role of Federal regulatory agencies

Over the years, several federal agencies like FDA, American Medical Association (AMA), Centers for Disease Control and Prevention (CDC) and the Institute for Safe Medication Practices (ISMP) have led a number of efforts to reduce the drug related accidents [1]. These include tracking drug overdose trends, creating education programs for health care providers and the public, introducing drug monitoring programs, highlighting important warnings [17], improving the TV commercials for the drug warnings [14] and, simplifying drug safety information with improved labeling. MedWatch service has been created by FDA for communicating the drug safety information in the form of alerts to the consumers [18, 19]. However, it requires active involvement from consumers like subscribing to the service, managing and searching the feeds, and is also not personalized to the patient's health situation. Therefore, a patient might not be receiving the warnings that apply to him/her.

#### 2.2 Mobile health (mHealth) apps

A number of mHealth mobile applications exist in the App stores with a potential to provide cost-efficient care delivery to patients and improve health outcomes. Smartphones/tablets form an ideal platform for mobile health solutions because of the ubiquity of these devices. Almost every mobile phone owner owns a smartphone today, a trend that is on the rise with already more than 140 million smartphone users in the U.S. [20]. The use of Android platform is preferred for the development of new mobile projects and its use has increased by nearly 96% each year since 2007, according to the data released by Black Duck Software [24, 25]. It provides the healthcare industry with an unprecedented growth opportunity. Mobile health app revenue is expected to grow tenfold by 2017, says a recently released report from research2guidance, a German market research firm [21]. Also, the number of mobile health apps published on the iOS and Android mobile operating systems have doubled in the past few years reaching 100,000 in the year 2014, based on an online survey of 2,000 mobile health app publishers and experts in the field [22, 23]. Also, the recent report on Mobile Health published by Pew Research Center says that one in three cell phone owners use their phones to look up health information. In addition, approximately 20% of smartphone/ tablet users have at least one health app on their device [26]. Health care organizations are trying to involve patients in their care by developing new mHealth apps and providing these tools to their patients to manage their health with a goal of improving it.

The mHealth applications available today on the Google play store and iTunes can be broadly categorized in *a*) *Health and Wellness Applications* that enable access to tools like calorie counters, prescription reminders, medical references, physician or

hospital locators, health news and information. For example, Mayo Clinic's Symptom Checker, WebMD, PocketPharmacist, etc [27, 28, 29, 30, 31]. b) Self-monitoring apps where the user tracks his or her routine/progress in an app. These are the apps that enable patients to control their own care, provide diet and exercise tracking assistance and thus, have the potential to help the patients lead a healthy life [32, 33, 34]. c) Chronic Disease Management apps are for patients with chronic health conditions such as Asthma, Alzheimer's disease, Diabetes, or heart disease and can help them adhere to their treatment regimens [35, 36]. Some apps are capable of recording the health measurements through the use of sensors with the smartphones like Glucometers for measuring blood glucose for Diabetic patients. d) Remote Monitoring apps offer remote monitoring linked to a smartphone app as a valuable, efficient and cost effective service provided by the healthcare providers to their patients [37]. e) Medical Reference apps are being used by the physicians throughout the health industry for a wide array of medical, and administrative tasks [16, 38, 39, 40, 41]. The most popular apps include medical reference apps, anatomical maps, drug information and educational apps. Other popular apps include AirStrip Cardiology [42], which can record patient's heart readings, and Epocrates [43], which functions as a research dictionary, prescription drug and safety information review tool, diagnostic lab test tool, disease guide, and drug-interaction checker. f) Personal Health Record apps provide patients an access to view/download their personal health information. These are generally coupled with an ability to schedule appointments, view test results and/or refill prescription medications [44].

An evaluation of mobile health applications tools conclude that a vast majority of the popular mobile health applications focus on wellness, fitness and self-monitoring.

The patient engagement offered by the current apps is limited and lack personalization and context awareness and thus, present an opportunity of improvement through the effectiveness of the technology. Patients expect a personalized advice based on their health history and circumstances, but the current applications fail to deliver this. Also, currently available solutions have a high signal fatigue in the form of alerts that needs to be addressed. Unnecessary warnings or overwarnings cause alert fatigue among patients and potentially influence their adoption of true contraindications.

#### 2.3 Drug Interaction Checkers

Several online systems exist for performing drug interaction checks for example WebMD, Drugs.com, rxlist.com, Walgreens.com, etc [45, 46, 47, 48]. However, these systems considerably lack in terms of considering personalization and demographic factors with respect to individual patient's conditions. Users need to explicitly mention the drug pair to check the interactions and thus, lack in user friendliness and are not very interactive.

#### 2.4 Clinical Decision Support Systems

A number of clinical decision support systems have been developed for avoiding adverse drug events and medication errors by physicians in a hospital setting [49, 50, 51, 52, 53]. However, the issues in designing patient-focused safe medication systems are very different and arguably harder because the users may have very little health literacy skills.

The use of knowledge-based expert systems (rule-based systems) [54, 55, 56, 57] that recommend or make decisions based on the knowledge gathered from domain experts has increased dramatically over the past few years. However, their use and

adoption in the pharmaceutical industry lags behind that of other fields. Pharmaceutical applications where their use has been proven successfully to reduce the cost of R&D, and save time during development processes are pharmaceutical formulations and preformulations [58, 59], drug tablet or capsule production [60], identifying and prioritizing drug candidates for clinical trials in humans [61]. Also, some of the other medical applications include helping in diagnostic processes, laboratory analysis, treatment protocol, and teaching and training of medical students and residents [62, 63, 64].

#### CHAPTER 3

#### EXPERT SYSTEMS

Knowledge-based expert systems or simply expert systems emulate the decisionmaking ability of a human expert for solving complex problems. These systems represent the expertise knowledge as data and a set of if-then rules [67, 68]. An expert system consists of two sub systems – knowledge base and inference engine. A knowledge base consists of rules and the data in the form of facts. The inference engine applies the rules to the facts to derive conclusions. The first rule-based expert system was developed in 1970s known as MYCIN [69], a program for medical diagnosis. The rule-based systems have seen a surge of interest in the past few years and there are a number of applications of rule-based systems in various domains [57, 70, 71, 72, 73, 74, 75]. Today, rule-based systems advise scientists, engineers, bankers, and are used in process planning, and design and manufacturing. These systems have started to gain popularity in the medical domain as well, with a number of applications as mentioned in Chapter 2. Developing such a system relieves the experts from the burden of always keeping up with and retaining large amounts of expertise knowledge, while ensuring accuracy and reliability of the decision-making.

#### 3.1 Architecture of a rule-based system

A typical rule engine as shown in Figure 1 comprises of 3 components – an inference engine, a working memory and a rule base [77, 78]. Working memory stores the data on which the rule engine will operate. The rule base contains all the rules of the

system. An inference engine functions as the primary component of a rule engine and controls the process of applying logical rules to the working memory to obtain the output. The inference engine consists of a pattern matcher, an agenda and an execution engine. A pattern matcher looks for all the rules in the rule base that match the facts in the working memory. Then, those rules are prioritized and their order of execution is determined and stored on the agenda. Finally, the execution engine component executes each of the matched rules' action part in the order determined. This cycle of matching rules, then selecting rules and finally executing them is repeated until no rules can be matched.

loop

match LHS conditions of the rules with facts in working memory if no rule matches then stop resolve conflicts (determine the order of execution) execute the action part of rule

end loop



Figure 1. The architecture of a rule-based system

Rule Engines can be broadly classified into 2 categories based on how the rules are scheduled for execution [76]. First, the *forward chaining* which is further categorized into two, a) the production/inference rule engines wherein rules are in the form of IF-THEN and are executed upon explicit invocation, and b) the reaction/event condition action rule engines where the rules are automatically invoked when events occur. Forward chaining starts with the available data and evaluates the rules till it reaches an answer and is often called data-driven approach. On the other hand, *backward chaining* is goal driven and thus, starts with a goal and tries to find any data that will lead to that goal. For performance enhancement, modern rule engines use Rete algorithm [79, 80] which is a pattern matching algorithm that remembers previous pattern matching results between iterations of the rule loop, and only updates the matches for the facts that change. For our application, the use of production rule engines [81, 82] fits the requirement of generating a safety advice for the user considering certain circumstances when the user explicitly initiates the request. Our application employs one of the fast rule-based system shells named Jess that uses enhanced and refined version of Rete Algorithm [78, 83].

#### 3.2 Jess Rule Engine

Jess is a powerful rule engine written in the Java language [83]. There are several benefits linked to using Jess such as stable Java API, flexible in use with command-line applications, GUI applications and servlets. One can create Java objects, call Java methods, manipulate Java objects, or in other words, can easily integrate with Java-based applications. Also, Jess is well supported and a stable software product and the upcoming version of Jess, i.e., Jess 8.0 includes native support for Android platform.

#### 3.2.1 Facts in the working memory

Jess provides three ways in which facts that hold data can be represented in the working memory, (i) Unordered facts (ii) Ordered facts and, (iii) Shadow facts [83]. An unordered fact is structurally similar to a row in a relational database table with its slots corresponding to table's columns. On the other hand, an ordered fact lacks the structure and thus, is similar to a flat list. Shadow facts are a form of unordered facts that represent Java objects and whose slots correspond to the properties of JavaBean classes. Jess automatically converts a JavaBean class to a fact's template, which has one slot for each JavaBean property. Various Jess functions are available to manipulate, add or remove facts from the working memory.

#### 3.2.2 Rules in Jess

All Jess rules [78, 83] are defined using the defrule construct and are uniquely identified by their names. Rules take actions based on the facts in the working memory. The rule consist of LHS that represent the IF conditional part (consists of patterns that can match facts) and RHS consisting of the action part (function calls; facts can be added to or retracted from the working memory). Jess offers various functions to explore the compiled Rete Network and to monitor the step-by-step execution. For example, "watch all" function prints all the diagnostics onto the screen during the execution, i.e., when rules are getting compiled, or facts are being asserted or retracted, and certain rules getting activated when their LHS conditions are met and finally firing of those activated rules. Rules can use constraints for constraining the slot data of facts, and conditional elements to express relationships between the facts (and, or, not, test etc.) [78].

#### CHAPTER 4

#### DESIGN OVERVIEW

Harnessing pervasive and cloud computing technologies can mitigate many factors contributing to high rates of pharmaceutical accidents. Pervasive devices (smartphones and tablets) are attractive platforms for building such a drug safety application because of their ubiquitous availability, computational and storage capabilities, and rich user interfaces. The purpose of this work is to design, develop and evaluate a proof-of-concept implementation of OTC drug safety advisory framework based on smartphone/tablet platform that is both personalized and context-aware in order to mitigate the pharmaceutical accidents. Our solution comprises of a mobile application hosted on the Android platform as a thin-client that consists of context and personalization centric components and, powered by the backend server for processing. The application intends to provide timely advice to a patient on whether the medicine he/she is considering to take is safe *under current circumstances*. The following sections explain various design factors that form the solution.

4.1 Design Features

The key to the system lies in three important factors, which are personalized, context-aware and user-friendly interface. The system is *personalized* because some people with certain conditions or demographics may be more sensitive to certain medications and the dosages, which are within the range of acceptable medical use, might be too much for their bodies to handle. Our solution takes into account patient's profile in

the form of health conditions and demographic characteristics while determining the safety of the drug the patient is intending to take. The solution is *context-aware* in the sense that it takes into account the medication intakes of the patient in the recent past while determining the safety advice. Thirdly, the aim is to develop *a simplified and user-friendly interface* such that patients with no prior experience with information technologies are capable of effectively using the app that is designed to help them take their medicine more safely. The solution takes into account the single active ingredient and multiple active ingredients drugs while generating the safety advice. This encompasses dosage calculations and contraindication checks taking into account the combination drugs that contain multiple active ingredients (AIs), and some AIs that are present in multiple marketed drugs.

#### 4.2 Functional Overview

Each patient interacts with the framework through his/her smartphone or tablet. After registering, the patient provides his/her profile by entering the patient specific information such as demography (age, gender, etc.), health conditions such as diabetes, hypertension, etc and the current prescription medications that the user is taking. Now, each time the user intends to take a new OTC drug, he/she provides the drug information to the system. Because this is a proof-of-concept application, the user selects one of the drugs that are currently configured in the application. Once the patient provides this information, the system performs safety determination and provides advice to the patient on whether it is safe to take the medication or not. The advice generated by the system can be categorized into three – no safety contraindication found, caution messages or the messages flagged as Unsafe. The advice also provide the reason why a particular drug is

unsafe to take, in the form of overdose, interaction with other drugs, unsafe age or due to a health condition. Example advice from the system include "Unsafe to take this drug as it may cause dangerous overdose of Acetaminophen" (for drugs such as Tylenol PM, Arthritis Pain Relief, etc.), "Unsafe to take Prilosec OTC with blood-thinning medicines such as Plavix" (for Prilosec OTC drug), or "No safety violations were found. However, please read the instructions carefully before taking the drug". The system also asks the user to confirm whether he/she is taking the drug. If the user confirms as "yes", the framework stores that drug information in the recent drug history, which will be used for future drug safety checks.

4.3 Drug Safety Criteria

This section highlights another important design feature that our solution supports, i.e., various drug safety criteria. First and foremost, *Temporal Drug overdosage* – the criterion that specifies the quantity of drug that should not be exceeded within a certain time period. *Demographic contraindications* – criterion specifying whether the drug is unsafe to take for patients of certain demographic characteristics (e.g., women, elderly, etc.). *Contraindications with respect to physiological state* – safety criterion pertaining to patients' health conditions (e.g., hypertension, heart disease, etc.). And lastly *Drug contraindications*, the criterion specifying the drugs that should not be used concomitantly.

#### 4.4 Safety Rules Design

Our application uses Jess rule language to develop the safety rules. Jess is very expressive and provides the power of implementing complex logical relations. The *first step* towards designing the rules is to gather the domain knowledge. The knowledge

acquisition is mainly done with an active involvement from the domain experts. For the proof-of-concept implementation, the domain knowledge was gathered from the trusted federal sources, such are FDA [86] and DailyMed [87, 88]. These repositories have up-to-date, high-quality information on most marketed drugs. The first step towards knowledge gathering is to search for the drugs and their active ingredients in these label databases. For example, Dologen OTC drug tablet consists of 2 active ingredients - Acetaminophen 650 mg and Dexbrompheniramine Maleate 2 mg, so FDA label database is searched for Dologen [89], Acetaminophen 650 mg [90] and Dexbrompheniramine Maleate 2 mg [91]. Then, we examined the safety warnings section of each of the above to understand the common parameters, division between the cautionary and unsafe categories, and the structure of facts and their slots before developing the safety rules. The *second step* in the process involves Knowledge Base Design, i.e., incorporating the acquired knowledge in the system and coming up with the required facts and the safety rules.

#### 4.4.1 Representing Facts and Slots

This step involves structuring the facts that would be required for invoking the actions of the rules and also, on which the rules can act. Based on the knowledge gathered from the first step, following are the facts and the slots created for the framework.

1. (User (userId) (userName) (userAge) (allergyList) (conditionList) (prescriptionList) (drugHistory))

2. (Medicine (name) (purpose) (constituentAI) (shortDetails))

3. (ActiveIngredient (name) (dose))

#### 4. (Advice (safetyCategory) (message))

User, Medicine, ActiveIngredient and Advice are the facts with their corresponding slots. The ActiveIngredient fact contains two slots, which are name and dose. The Dose slot holds the quantity and the units of measurement for the given active ingredient. Often times, there are multiple active ingredients present in a single drug and thus, the number of ActiveIngredient facts in the working memory will grow with that count. The Medicine fact represents the medicine that the user intends to take, and holds the name of the OTC medicine, the purpose of the medicine e.g. Pain Reliever, fever reducer etc, list of all the active ingredients present and a short description. The User fact contains information pertaining to the patient and holds demographic slots like age, etc. along with the list of allergies, health conditions, and prescription drugs. User fact also consists of a drugHistory slot that keeps the temporal record of all the previous medications taken in the recent past. Lastly, Advice fact holds the generated advice for the patient. It is part of the RHS (Action) of a safety rule and thus, gets added to the working memory once a rule is fired. Its slot named safetyCategory represents the safety message category, i.e., UNSAFE or CAUTION, and the message slot holds the actual advice message.

As explained earlier, shadow facts in Jess are the facts that represent JavaBean objects with the slots representing the properties of a JavaBean class. The templates for these facts (similar to schema) can be created once the JavaBean class is supplied to the deftemplate function of Jess, i.e., using (deftemplate User (declare (from-class User))). Another way is by using defclass function that takes care of creating the templates for the facts internally. Our implementation uses shadow facts for representing the factual data of the system through various Java Bean classes, which are User.java, Advice.java,

Medicine.java, etc. The slots get automatically populated with the values of the JavaBean properties once the link is established.

#### 4.4.2 Safety Rules

Developing the production rules involves writing the precondition (IF) and an action (THEN) part. A Jess rule looks like LHS => RHS where LHS holds the patterns that are matched to the facts in the working memory and the RHS holds the function calls. For preconditions, the safety rules for our application can be divided into two levels a) medicine level, and b) active ingredient level. *Medicine level* rules are the rules that apply at a high level or the ones that are repeated amongst multiple active ingredients of the medicine. From the analysis step, it was observed that certain safety warnings exist only in the medicine label and not in its active ingredients' labels while others exist in both the medicine labels and the active ingredients labels (with same or different values) thus making the same warnings repeat a number of times while authoring the rules separately. A perfect example of the second case is age restriction warning that says that patients under certain age should not take this drug or, alcohol warnings that caution the user against consuming alcohol with the drug. On the other hand, active ingredient level holds the rules representing the safety warnings that apply to that particular active ingredient. This design decision is particularly significant because it removes duplicates and promotes reusability. For example, current implementation of the application includes the rules for 10 OTC drugs and active ingredient named Acetaminophen is present in 6 of those but the rules for Acetaminophen lie in a single file, which is referred by all of the 6 OTC drugs. The RHS (THEN part) of the safety rules add a new fact into the working memory named Advice. Once the run command is issued to the Jess, it starts

firing the activated rules (rules whose conditions evaluate to true) and executes the actions on their RHS. After execution of the rules, our application fetches all the Advice facts from the working memory and displays to the patient.

For illustrating all of the above points, let us look at the Dologen OTC drug example. Three separate rule files are created for representing the safety warnings for Dologen which are: Dologen.plsf, Acetaminophen.plsf and Dexbrompheniramine Maleate.plsf. Dologen.plsf consists of alcohol warning rule, pregnancy rule, Age restriction rule, and several health conditions rules that apply at the medicine level. Below are the two sample rules from Dologen.plsf:

1. (defrule Dologen-age-restriction

"Restricted age for Dologen administration"

(User {userAge < 12})

=>

(add (new Advice "Unsafe Age: Do not use Dologen in children under 12 years of age.")))

2. (defrule Dologen-alcohol

"Alcohol warning"

=>

(add (new Advice (SafetyStatus.CAUTION) "Severe liver damage may occur if you take 3 or more alcoholic drinks every day while using Dologen.")))

The first rule named Dologen-age-restriction supports the *Demographic* 

contraindication drug safety criteria mentioned earlier and checks whether patient's age

is less than 12 and if Jess finds that pattern, the rule's RHS adds the Advice object (Jess internally creates a shadow fact) into the working memory. The second rule named Dologen-alcohol is a cautionary rule and as you can notice, it does not contain any LHS meaning that this rule should always be executed and thus, this advice should always be displayed to the user. Similarly, the rules pertaining to each of the active ingredients are added into the respective rule files. As an example, few of the Acetaminophen rules are shown below:

1. (defrule Acetaminophen-overdosage

"Acetaminophen overdose"

(User (OBJECT ?us))

(test (?us exceedsDrugDoseInWindowHours "Acetaminophen" 4000 24))
=>

(printout t "Acetaminophen overdose for " ?us.userName crlf)
(add (new Advice "Unsafe to take this drug as it may cause dangerous
overdose of Acetaminophen"))

2. (defrule Acetaminophen-allergy

"Allergic reaction to Acetaminophen"

(User (allergyList ?allergies))

(test (?allergies contains "Acetaminophen"))

=>

(add (new Advice "Unsafe as allergic to Acetaminophen (AI).")))

The first rule named Acetaminophen-overdosage checks for the overdosage of the Acetaminophen. This is the rule that represents *temporal-aware drug overdosage* drug

safety criteria. It checks whether Acetaminophen's total quantity for the user exceeds 4000 mg in the last 24 hours. This rule calls a Java method named exceedsDrugDoseInWindowHours and passes Acetaminophen, 4000 and 24 as arguments, which in turn checks the drug history of the user along with the new medicine supplied and calculates the total dosage of the active ingredient within the specified time period. Our framework verifies the strength of the active ingredient while calculating the total dosage; this is particularly important because the active ingredient's strengths vary with each drug (e.g Dologen contains Acetaminophen, 650 mg whereas Tylenol PM contains Acetaminophen, 500 mg). The second rule named Acetaminophen-allergy belongs to the drug safety criteria - *Contraindications with respect to physiological state* and goes through the allergy list supplied by the patient to check if he/she is allergic to Acetaminophen. The full list of rules for the Dologen OTC drug has been provided in the Appendix B.

#### CHAPTER 5

#### SYSTEM ARCHITECTURE

Figure 2 depicts the architecture diagram of our application. The solution follows the client-server architecture connected via network where mobile application hosted on Android platform acts as a thin client and the backend server hosts the Jess rule engine, which, after executing the rules, serves the client with the generated personalized advice.



Figure 2. The System Architecture Diagram

The proposed system has been implemented using Java programming language. The front-end app has been developed using Android SDK [92] and acts as a means of sending the user inputs to the backend, receiving the advice response and presenting it to the patient using its display capabilities. The backend is implemented as a Java-based RESTful web service that processes the HTTP POST requests containing context and personalization information in JSON format, sent from the Android app. After processing the request, the web service responds back with the generated advice again using JSON as a data-interchange format. As mentioned earlier, the patient creates a profile in the Android app after registration. User profile is stored locally in the user profile database, i.e., on the mobile device (SQLite). A possible extension to the system could be extracting this information from patient's electronic health record. The drug history list stores the information regarding the recent medicines taken by the patient, i.e., the name of each drug, quantity and the time at which it was taken. The drug history gets populated when the patient confirms taking a drug. While a patient's profile is key to personalization, recent drug history is central to achieving the context-awareness for the framework. Following subsections explain various modules of the overall solution. 5.1 Active ingredient retrieval module

The active ingredient information retrieval module (AI retrieval module) is responsible for discovering the medicines that the patient is considering to take and, their active ingredients along with the strengths and dose quantities for the drugs. This is needed for ensuring accurate safety checks, especially for combination drugs. The module is hosted on the server side and acts as an entry point for the requests from the Android app. It works by first accepting the HTTP POST request, fetching the user and the OTC drug information from it, and then extracting all the necessary inputs required for processing by other system components. This module also takes care of converting the JSON format input data into respective Java objects for processing by the application and adding the User, Medicine and Drug facts into the Jess working memory. Also, for each OTC drug, safety-warning rules are separated in different rule files (one for OTC drug and one for each AI) and are stored in the Rules Repository based on the class that

the rules file falls in, i.e., medicine or active ingredient. The respective rule files are pulled from the repository in an ad-hoc manner depending upon the context attributes and the drug that the user is considering to take. These rules are then loaded into the rule engine for execution and performing the drug safety checks.

#### 5.2 Multi-criteria drug-safety rule engine

The multi-criteria drug safety rule engine (MCDS rule engine) forms an important module of the framework. The MCDS rule-engine is responsible for performing the safety checks for each active ingredient in the given drug. This is accomplished by executing various drug safety rules authored for that drug and its active ingredients. As explained earlier in the design section, Jess is being used as a rule engine and for coding the drug safety rules. After the AI retrieval module populates Jess's working memory and supplies the rules, application's MCDS rule engine starts applying the rules to the facts in working memory to obtain the output advice. The generated advice fact(s) are pulled from the working memory once the execution is complete and passed onto the Advice convertor module for preparing the response.

#### 5.3 Advice Convertor

This component resides on both the client side (Android app) and the server side. At the server side, it is responsible for getting the response advice from the rule engine and converting to JSON format before sending the response to the client. At the client side, the component accepts the response and converts into Java objects (Advice) for displaying it to the end user in appropriate alert boxes based on the category of the advice (CAUTION, UNSAFE or no safety concerns found). Our application uses GSON library for converting JSON to Java objects and vice-versa [93].

#### 5.4 The User Interface

An Android app acts as a user engagement tool through a simple front-end interface and serves to record the personalization and context aware information from the patient. It offers *Register Screen* as shown in Figure 3 for registration by the first time users and Login screen shown in Figure 4 for login into the app.

🔗 Register Profile	
ull Name	
john bach	
mail	
john@aol.com	
assword	
Create profile	
Already have an account? Login he	re!

Email john@a	ol.com
Password	
	Login
	PillSafe? Create your profile he
New to	

Figure 3. Register Screen

Figure 4. Login Screen

There are three main screens that play the most important role in collecting all the patient information a) *User profile screen* as shown in Figure 5 collects the demographic characteristics, health conditions and prescription medications. User profile screen provides full control to the patient to be able to edit the profile, add new entries or delete the existing entries from the profile. It also shows user's drug history in the recent past. b) *Safety Checker Screen* (Figure 7) is the one that initiates the safety advice request to the backend. It takes the medicine input from the patient that he/she is considering taking

along with its quantity and passes this information along with patient's profile to the backend web service.

nelcome t	Q :	
PROFILE	SAFETY CHECKER	MEDICINE INF
$\sim$ Account Deta	ails	1810
Allergies		AND
Health Condi V	tions	AND
Prescription	Drugs	A. 1810
clopidogrel		
Administratic	on History	AMIN

Figure 5. User profile Screen

🎸 Welcome to PillSafe 🛛 🔍 🚦							
TIME ADJ. SAFETY CHECKER MEDICINE IN							
Select the Medicine:							
Tylenol S	Sinus Congestion	and Pain					
Tylenol ES							
Quantity:							
1							
2							
Safety Check							

Figure 7. Safety Checker

		36	10:19
🔗 Welcome	to PillSafe	Q,	i
PROFILE	SAFETY CHECKER	MEDI	CINE INFO
Edit Accou	nt details		2
Full Name	1		
Email	l.com		
Age 22			_ •
Cance	I	Save	

Figure 6. Edit Account



Figure 8. CAUTION message

Lastly, c) *Advice Dialog box* presents the safety advice generated by the backend to the patient. The safety advice response consists of the category of the advice and an actual message. Our Android app differentiates between various advice based on the safety category supplied in the response. If all of the messages generated are of category "CAUTION", then cautionary alert box pops up (Figure 8) with all the advice messages in it. On the other hand, if even one of the returned advice is of category "UNSAFE", then all of the advice messages including the cautionary ones are presented in the unsafe alert box as shown in Figure 10. If none of the safety violations/cautionary messages are found by the backend web service, then "No safety violations found" message is presented to the user (Figure 9).



Figure 9. No safety concerns



Figure 10. UNSAFE message

#### CHAPTER 6

#### **EVALUATION**

The evaluation covers both the qualitative and quantitative analysis. The quantitative experiments were performed in order to measure 2 types of response times that our system takes in generating the advice, a) the *end-end system response time* which measures the time taken by the request from the Android device till the response is received back from the backend, b) the *rule engine response time* measures the time taken by the Jess Rule engine in order to execute the rules and generate the advice. The 10 OTC Drugs chosen for experiments are a combination of single active ingredient and multiple active ingredients and the total number of rules in our system are 122 with the number of rules written for each drug varying between 7 and 18. Please refer to Appendix A for a complete list of OTC drugs along with their active ingredients and the total number of rules and the facts created in our application. All the tests were conducted with Linux OS on the same machine with characteristics: Intel(R) Core(TM) i7-3770 CPU, 3.40GHz, 8GB RAM, Linux Fedora, 64-bit. The rule engine was 'warmed up' by executing the rule sets few times and then an average elapsed time was measured for 10 execution cycles. Following subsections explain various aspects of the evaluation. 6.1 Safety Warnings in textual form vs. Java code vs. Jess Rules

This evaluation forms the qualitative analysis and compares the textual safety warnings on the drug labels with the warnings represented in Java language and finally, with the safety rules authored in Jess. a) *Safety warnings written on the drug labels* are

usually in the long textual form. Following is a snippet of warnings for OTC drug named Dologen taken from the FDA label repository [89].

"Liver Warning: This product contains acetaminophen. Severe liver damage may occur if you take: more than 6 tablets in 24 hours, with other drugs containing acetaminophen (prescription or non prescription). Ask a doctor or pharmacist before using with other drugs if you are not sure. Ask a doctor before use if you have heart disease, high blood pressure, diabetes or thyroid disease. Children under 12 years of age: Consult a doctor."

FDA Label Database contains around 66,000 drug labels and uses SPL (structured product labeling) markup format for representing the label information [94]. The above is just a small section of the lengthy warnings. Reading the long and lengthy warnings, understanding and extracting the information that is relevant to the patient is a tedious task especially if the patient has complicated health conditions (e.g., multiple chronic illnesses), or takes multiple prescription/OTC medications. Patients may easily miss an important safety warning or may forget the past medications and their dosages. b) Safety warnings can be *encoded in the Java application* though it would involve significant conditional branching. However, the dynamic nature of the safety warnings makes this approach less manageable and less maintainable because changes in the warnings would entail modifying the application code repeatedly. Existing safety warnings may change over the time and new warnings may be added for the drugs hence, making this approach unsuitable for our framework, especially with thousands of OTC drugs already in the market. c) Lastly, rules written in Jess separates the repository of domain knowledge that keeps on evolving over the years, from the rest of the application. Thus, separating the

business logic in rules makes it easy to introduce new changes to the application thus, making it more maintainable. Knowledge with the rules is easy to express and understand because of their uniformity and simplicity. Please refer to Appendix B for the equivalent rules for the earlier textual snippet of the safety warnings for Dologen. Our application takes care of generating a personalized advice for the OTC medication taking into account patient's profile and the drug history while executing the safety rules. This means it extracts and presents the drug label information to the patient that is relevant to him/her sparing the patient from the tedious task of going through the long list of warnings on the drug label and hence reducing the risk of medication errors. The outcome of the safety rules execution was along the expected lines for each of the test cases (i.e., there were no errors in the rule executions).



Figure 11. Rete Network for Dologen safety rules

In terms of performance, the Jess rule engine uses an enhanced form of Rete algorithm for optimized pattern matching. Figure 11 shows the Rete network generated for Dologen safety rules. Facts enter from the top of the network containing single input nodes and are sent down while going through a series of tests to reach the terminal nodes that represent the individual rules. The path from the root node till the terminal node represents a rule's entire LHS. When a set of facts reaches a terminal node, an activation record is created for that rule and the corresponding facts. Performance enhancement is achieved first by sharing of the nodes between the rules in the Rete Network, thus avoiding any duplicate computations. Moreover, Rete algorithm remembers past iteration's test results and thus, pattern matching is performed only for the changes in the working memory and not for all the facts during the next iterations. Also, Jess makes an attractive choice for the implementation of our application because of some of the independent performance benchmarks [84, 85, 97]. E. J. Friedman-Hill mentions in his book "Jess in Action" [78] that Jess is capable of firing more than 80,000 rules per second, can add more than 100,000 facts to the working memory per second and can perform nearly 600,000 pattern matching operations per second.

#### 6.2 Rule Engine Response time

This category of performance evaluation measures the time taken by the Jess rule engine to process the information and execute the rules to return the advice. The rule engine response time is further measured for two types of User facts, first, containing empty slots and second, with all of the slots filled with values. Where the first case fires only the cautionary rules, i.e., minimum number from the given rule set, the second case is a deliberate effort to make maximum of the total rules fired for a given OTC drug. We perform two types of tests for measuring the rule engine response time. For the first experiment, the rule engine response time measurement involves a) loading of the rules, b) compiling those to form the Rete Network, c) assertion of the facts into the working memory, d) running the engine to execute the rules and finally, e) extracting the Advice facts from the working memory. Table 1 shows the average response times taken by the rule engine for performing all of these 5 steps for both the empty User fact and the filled

User fact. The results show that the elapsed time increases as the number of rules increase. OTC drug named Motrin IB has the highest number of rules and takes highest time to execute (both for an empty User fact and filled User fact). The table also shows the comparison between the elapsed times for the empty User fact and the filled User fact and it is observed that the elapsed time is more in the case of filled User fact for each OTC drug. This is mainly because of the additional processing done by the rule engine in terms of more rule activations and executions for the filled User fact.

Medicine	Total number	Number of	Number of	Average	Number of	Average
name	of Rules	facts added to the working	Rules fired (Empty User fact)	elapsed time in ms (Empty User fact)	Rules fired (Filled User fact)	elapsed time in ms (Filled User fact)
		memory				
Tylenol Sinus	13	5	1	8	12	9
Congestion						
and Pain						
Severe						
Tylenol ES	7	3	1	5	6	5
Tylenol PM	12	4	2	7	11	9
Robitussin	14	5	2	8	13	10
Daytime Cold						
and Flu						
Arthritis Pain	7	3	1	4	6	5
Relief						
Prilosec OTC	12	3	0	6	12	8
Motrin IB	18	3	1	9	17	11
Dologen	15	4	2	8	14	10
All Day Pain	16	3	1	8	15	10
Relief						
Sudafed	8	3	0	5	8	5

Table 1. Rule Engine Response Time (Experiment I)

Now, the second set of tests is targeted towards finding which of the earlier mentioned five steps of the rule engine execution takes a lot of time and if there is a way of sharing that step amongst multiple client requests in order to improve the performance. We excluded the times for the first two steps, i.e., loading of the rules and compiling those to form the Rete Network, and instead measured the elapsed time for the rest of the steps, i.e., a) assertion of the facts into the working memory, b) running the engine to execute the rules and finally, c) extracting the Advice facts from the working memory. Table 2 represents the average elapsed times recorded in this experiment as compared to the first experiment. We noticed a considerable decrease (on average, 40%) in the response times as compared to the first test (for both empty user fact and filled user fact). So, sharing of the first two steps between the client requests can provide a significant performance improvement. In order to accomplish this, Jess offers peering of the rule engine using which the compiled rules can be shared amongst various client requests.

	Experi	iment I	Experiment II			
Medicine name	Average elapsed time in ms (Empty User fact)	Average elapsed time in ms (Filled User fact)	Average elapsed time in ms (Empty User fact)	Average elapsed time in ms (Filled User fact)		
Tylenol Sinus Congestion and Pain Severe	8	9	3	5		
Tylenol ES	5	5	1	3		
Tylenol PM	7	9	2	5		
Robitussin Daytime Cold and Flu	8	10	4	5		
Arthritis Pain Relief	4	5	1	3		
Prilosec OTC	6	8	2	5		
Motrin IB	9	11	2	7		
Dologen	8	10	2	5		
All Day Pain Relief	8	10	2	6		
Sudafed	5	5	1	4		

Table 2. Rule Engine Response Time (Experiment I vs. Experiment II)

#### 6.3 End-to-end Response time

In this evaluation, end-to-end response time is measured from the Android app running on a mobile device. The elapsed time includes the network overhead, i.e., establishing the connection with the backbone network, transmitting input data, and finally receiving the response back from the backend server. The Android device used for this experiment is LG Nexus 4 with Android version: 4.4.4, 2GB RAM and CPU as Dualcore 1.5 GHz ARM. The elapsed time is measured for both the mobile network and the Wi-Fi network. In case of Wi-Fi, the first test was performed in the home Wi-Fi network where the backend server was also located in the same network and, the second test was performed in the public Wi-Fi network where our backend server was located outside of the network. The experiments included measuring the end-to-end response times 12 times and then dropping the highest and the lowest values and calculating the average elapsed time with the rest of 10 samples. Table 3 shows the end-to-end response times measured from the mobile device for three networks. This experiment gives us an idea of the response times that can be expected in real world scenario in the presence of the network overhead.

Tab	le 3.	End	l-to-	end	Res	ponse	Time
-----	-------	-----	-------	-----	-----	-------	------

Network Type	Ping time in ms (hitting the	Average end-end elapsed time in ms	Average end-end elapsed time in ms
	backend server)	(Empty User Fact)	(Filled User Fact)
Home Wi-Fi	15	42	66
Public Wi-Fi	45	268	302
T- Mobile (3G)	450	1223	1883

The results again assert that the elapsed time increases in the case of filled User fact as compared to an empty User fact. Also, the results show that communication over the mobile network involves a higher network overhead in comparison to the Wi-Fi network.

#### CHAPTER 7

#### CONCLUSION

This research project designs and develops a rule-engine-based implementation of OTC drug-safety advisory application that generates timely safety advice cautioning its user about the medicine that he/she is considering to take. The key benefits of this framework are served by three powerful features, which are personalized, context-aware and user-friendly interface. The framework supports four main OTC drug safety criteria, which are drug overdosage, drug contraindications, demographic contraindications and physiological state contraindications. The evaluation results demonstrate the effectiveness of our approach and the efficiency of our application.

7.1 Significance and broader impact

Our application aims at overcoming the limitations of current systems and empowering the patients and caregivers in making safe medication decisions with regards to the OTC drug intakes. The OTC drug safety application will particularly benefit the populations that are most vulnerable to drug related accidents such as the elderly, rural populations, and non-native speakers.

#### 7.2 Future work

Future work in the design includes support for another drug safety criteria, i.e., Drug-food/beverage contraindications. The current implementation handles this as a part of the cautionary advice. Secondly, the system would include the support for the drug category/purpose that will help in expressing the safety warnings that depend on these

like "Ask a doctor or pharmacist before use if you are taking *prescription antifungal or anti-yeast* medicines".

One of the significant changes to the system architecture would be to migrate the MCDS rule engine from cloud to the Android app. Jess 8.0, which will be released in the near future, intends to provide native support for Android [95]. This would bring huge performance boost in terms of responsiveness and will ease the dependency of the framework on the network availability as the data could be cached locally on the mobile device. Because all of the rules need to be manually authored, this proof-of-concept framework includes the rules for 10 OTC drugs. Future work also includes support for more OTC drugs and devising means for automatic extraction of the warnings and formulation of the safety rules.

#### REFERENCES

[1] Opioids drive continued increase in drug overdose deaths

http://www.cdc.gov/media/releases/2013/p0220\_drug\_overdose\_deaths.html

[2] 100 Americans die of drug overdoses each day. How do we stop that?

http://www.washingtonpost.com/blogs/wonkblog/wp/2014/02/07/100-americans-die-of-

drug-overdoses-each-day-how-do-we-stop-that/

[3] Prescription Painkiller Overdoses,

http://www.cdc.gov/vitalsigns/prescriptionpainkilleroverdoses/

[4] Drug Overdose Deaths up for 11th Straight Year (USA Today Article).

http://www.usatoday.com/story/news/nation/2013/02/19/drug-overdose-deaths/1931261/

[5] Over-the-counter analgesics in older adults: a call for improved labelling and

consumer education, Drugs & Aging, 2004; 21(8):485-98

[6] Common medications with Acetaminophen,

http://www.knowyourdose.org/common-medications

[7] Drug development and drug interactions

http://www.fda.gov/Drugs/DevelopmentApprovalProcess/DevelopmentResources/DrugIn teractionsLabeling/ucm080499.htm

[8] Concomitant Use of Ibuprofen and Aspirin: Potential for Attenuation of the

AntiPlatelet Effect of Aspirin

http://www.fda.gov/downloads/Drugs/DrugSafety/PostmarketDrugSafetyInformationforP atientsandProviders/UCM161282.pdf

[9] Food-Drug Interactions, http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3191675/

[10] Precautions with Over-the-counter Drugs,

http://www.merckmanuals.com/home/drugs/over-the-

counter\_drugs/precautions\_with\_over-the-counter\_drugs.html#v716888

[11] Aging and Drugs,

http://www.merckmanuals.com/home/older\_peoples\_health\_issues/aging\_and\_drugs/agin g\_and\_drugs.html

[12] Risk factors for Adverse Drug Reactions

http://www.merckmanuals.com/home/drugs/adverse\_drug\_reactions/risk\_factors\_for\_adv erse\_drug\_reactions.html

[13] How many Drugs are you taking?

http://www.webmd.com/healthy-aging/features/how-many-drugs-are-you-taking

[14] FDA looks into limiting TV commercials' rambling lists of drug risks

http://www.nydailynews.com/news/national/fda-studies-limiting-tv-ads-drug-risk-lists-

article-1.1617940

[15] Drug Safety and Availability, http://www.fda.gov/Drugs/DrugSafety/default.htm

[16] Docs avoid drug errors with mobile apps.

http://www.healthcareitnews.com/news/docs-avoid-drug-errors-mobile-apps

[17] New Tylenol cap will have warning label,

http://www.cnn.com/2013/08/29/health/tylenol-cap-warning/index.html?hpt=hp\_t2

[18] Subscribe to MedWatch Safety Alerts

http://www.fda.gov/Safety/MedWatch/

[19] FDA's Safe Use Initiative - Collaborating to Reduce Preventable Harm from Medications, 11 2009.

[20] Smartphone, Tablet Uptake Still Climbing in US,

http://www.emarketer.com/Article/Smartphone-Tablet-Uptake-Still-Climbing-

US/1010297

[21] research2guidance's mHealth App Developer Economics 2014,

http://research2guidance.com/r2g/mHealth-App-Developer-Economics-2014.pdf

[22] Mobile health app revenue to grow tenfold by 2017, study predicts

http://www.modernhealthcare.com/article/20140522/BLOG/305229997

[23] Mobile Applications in Healthcare http://divurgent.com/images/mHealthcare.pdf

[24] Current State of Android Mobile Apps and mHealth Projects

http://www.openhealthnews.com/hotnews/current-state-android-mobile-apps-and-

mhealth-projects

[25] Cummulative Open Source Projects Graph

http://www.blackducksoftware.com/files/images/Cumulative-Open-Source-Mobile-

Projects.png

[26] Pew Report: Mobile Health 2012

http://www.openhealthnews.com/resources/pew-report-mobile-health-2012

[27] WebMD, https://play.google.com/store/apps/details?id=com.webmd.android

[28] Healthful Reminders for Medications, Beyond an Apple a Day

http://www.nytimes.com/2010/09/30/technology/personaltech/30smart.html?\_r=1&

[29] My Pills - a Pill reminder app,

https://play.google.com/store/apps/details?id=simo.mycure&hl=en

[30] MediSafe Meds and Pill Reminder,

https://play.google.com/store/apps/details?id=com.medisafe.android.client&hl=en

[31] PocketPharmacist - Drug Information, Interaction Checker, and Medication Organizer.

https://itunes.apple.com/us/app/pocketpharmacist-drug-information/id387365379?mt=8 [32] Fitbit Mobile,

https://play.google.com/store/apps/details?id=com.fitbit.FitbitMobile&hl=en

[33] Calorie Counter – MyFitnessPal,

https://play.google.com/store/apps/details?id=com.myfitnesspal.android

[34] ShopWell – Healthy Diet and Grocery Food Scanner

https://itunes.apple.com/us/app/healthy-diet-grocery-food/id393422300?mt=8

[35] Diabetes Manager,

https://play.google.com/store/apps/details?id=com.welldoc.diabetesmanager

[36] Asthma Monitor,

https://play.google.com/store/apps/details?id=ch.imvs.unibas.asthma

[37] Mayo Clinic, https://play.google.com/store/apps/details?id=com.mayoclinic.patient

[38] Medscape from WebMD,

https://play.google.com/store/apps/details?id=com.medscape.android

[39] 9 Mobile Health Apps worth a closer look,

http://www.informationweek.com/mobile/9-mobile-health-apps-worth-a-closer-look/d/d-

id/1101548?itc=edit\_in\_body\_cross

[40] Apps, Doctors and Digital Devices

http://industryreport.jacksoncoker.com//physician-career-

resources/newsletters/monthlymain/des/Apps.aspx

[41] Mobile Medical Apps,

http://www.informationweek.com/mobile/12-hot-mobile-medical-apps/d/d-

id/1097466?page\_number=7

[42] AirStrip – Cardiology, https://play.google.com/store/apps/details?id=com.ast.cardio

[43] Epocrates, https://play.google.com/store/apps/details?id=com.epocrates

[44] Top 4 apps for Organizing Your Medical Records

http://www.healthedeals.com/articles/4-top-apps-for-organizing-your-medical-records

[45] Drugs.com – Drug Interaction Checker,

http://www.drugs.com/drug\_interactions.html

[46] WebMD Interaction Checker, http://www.webmd.com/interaction-checker/

[47] Walgreens - Check Drug Interactions,

https://www.walgreens.com/pharmacy/library/checkdrug/selectfirstdrug.jsp

[48] RxList – Drug Interactions Checker, http://www.rxlist.com/drug-interaction-

checker.htm

[49] J. G. Anderson. Clearing the way for physicians' use of clinical information systems.Commun. ACM, 40(8): 83–90, Aug. 1997.

[50] Y. Zhang, S. Fong, J. Fiaidhi, and S. Mohammed. Real-time clinical decision support system with data stream mining. J. Biomed. Biotechnol., 2012:580186, 2012.
[51] I. K. Bindoff, P. C. Tenni, G. M. Peterson, B. H. Kang, and S. L. Jackson. Development of an intelligent decision support system for medication review. J Clin Pharm Ther, 32(1):81–88, Feb 2007.

[52] Y. Bttiger, K. Laine, M. Andersson, T. Korhonen, B. Molin, M.-L. Ovesj, T.

Tirkkonen, A. Rane, L. Gustafsson, and B. Eiermann. Sfinxa drug-drug interaction database designed for clinical decision support systems. European Journal of Clinical Pharmacology, 65(6):627–633, 2009.

[53] Mohamed Ramadan and Khalid Al-Saleh. Development of an Expert System for reducing Medical Errors. IJSEA, Vol.4, No.6, November 2013

[54] Some Guidelines For Deciding Whether To Use A Rules Engine

http://www.jessrules.com/guidelines.shtml

[55] A. Polleres. Proceedings of the Third International Conference on Web reasoning and Rule Systems. Springer, Berlin, 2009.

[56] C. L. Forgy. Expert systems. chapter Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, pages 324–341. IEEE Computer Society Press, Los Alamitos, CA, USA, 1990.

[57] E. J. Friedman-Hill. Jess: The Rule Engine for the Java Platform. Technical report, Sandia National Laboratories, 2008.

[58] Expert System In The Development Of Pharmaceutical Formulations

http://www.pharmainfo.net/reviews/expert-system-development-pharmaceuticalformulations

[59] KnowledgeBench, http://en.wikipedia.org/wiki/KnowledgeBench

[60] Nopphadol Chalortham, Phuriwat Leesawat, Marut Buranarach and Thepchai

Supnithi, Ontology Development for Pharmaceutical Tablet Production Expert System,

Proceedings of ECTI-CON, 2008.

[61] Expert Systems and Knowledge Bases for pharmaceutical research

http://www.pharmainformatic.com/PharmaInformatic\_englisch.pdf

[62] Santhiyakumari, N., Rajendran, P. & Madheswaran, M. (2011) "Medical Decision-Making System of Ultrasound Carotid Artery Intima–Media Thickness Using Neural Networks" Journal of Digital Imaging, Vol.24, pp.1112–1125

[63] Tomar, P.P., Singh, R. & Saxena, P.K. (2012) "A Medical Multimedia based

Clinical Decision Support System for Operational Chronic Lung Diseases Diagnosis and

Training", International Journal of Computer Applications, Volume 49, No.8, pp. 1-12.

[64] Munirah, M., Rozlini, S.F., Chew, (2012) Design and Rules Development of Online

Children Skin Diseases Diagnosis System, International Conference on Information and

Knowledge Management (ICIKM 2012) IPCSIT vol.45, pp.272-276.

[65] Drug Poisoning Deaths in the United States, 1980–2008

http://www.cdc.gov/nchs/data/databriefs/db81.pdf

[66] Prescription Drug Overdose in the United States: Fact Sheet

http://www.cdc.gov/homeandrecreationalsafety/overdose/facts.html

[67] Expert Systems, http://www.pcai.com/web/ai\_info/expert\_systems.html

[68] Expert Systems, http://en.wikipedia.org/wiki/Expert\_system

[69] MYCIN, http://en.wikipedia.org/wiki/Mycin

[70] E. N. Hanson. The design and implementation of the ariel active database rule system. IEEE Transactions on Knowledge and Data Engineering, 8(1), 1996

[71] A. Polleres. Proceedings of the Third International Conference on Web reasoning and Rule Systems. Springer, Berlin, 2009.

[72] Eric Brill, A simple rule-based part of speech tagger, Proceedings of the workshop on Speech and Natural Language, 1992 [73] Chen, Yen-Lin; Hong, Zeng-Wei; Chuang, Cheng-Hung "A knowledge-based system for extracting text-lines from mixed and overlapping text/graphics compound document images", Expert Systems with Applications, Volume.39, Issue.1, pp.494, 2012, ISSN: 09574174

[74] Cucchiara, R, Piccardi, M; Mello, P. Image analysis and rule-based reasoning for a traffic monitoring system. Intelligent Transportation Systems, IEEE Transactions, 2000.
[75] Bernard, J.A. Use of a rule-based system for process control, Control Systems Magazine, IEEE, 1988

[76] Business Rules Engine, http://en.wikipedia.org/wiki/Business\_rules\_engine

[77] Rule based system, http://en.wikipedia.org/wiki/Rule-based\_system

[78] Ernest Friedman Hill, Jess in Action: Java Rule-Based Systems, Manning Publications Co., Greenwich, CT, 2003

[79] C. L. Forgy. Expert systems. chapter Rete: A Fast Algorithm for the Many
Pattern/Many Object Pattern Match Problem, pages 324–341. IEEE Computer Society
Press, Los Alamitos, CA, USA, 1990.

[80] The Rete Algorithm, http://www.jessrules.com/doc/71/rete.html

[81] Production System, http://en.wikipedia.org/wiki/Production\_system

[82] Forward Chaining, http://en.wikipedia.org/wiki/Forward\_chaining

[83] F. E.J. Jess, the rule engine for java platform.

http://herzberg.ca.sandia.gov/jess/docs/71, version7.1, 2008

[84] C. Young, Microsoft's rule engine scalability results - a comparison with jess and

drools, 2004. http://geekswithblogs.net/cyoung/articles/54022.aspx

[85] Senlin Liang, Paul Fodor, Hui Wan, Michael Kifer. OpenRuleBench: An Analysis of the Performance of Rule Engines, WWW, 2009.

[86] FDA Labeling, https://open.fda.gov/update/drug-product-labeling/,

http://labels.fda.gov/

[87] NLM, http://druginfo.nlm.nih.gov/drugportal/drugportal.jsp

[88] DailyMed, http://dailymed.nlm.nih.gov/dailymed/index.cfm

[89] Dologen,

http://www.accessdata.fda.gov/spl/data/6abb539d-677d-4ad9-9637-

f8ed5bbeb58e/6abb539d-677d-4ad9-9637-f8ed5bbeb58e.xml

[90] Acetaminophen 650 mg,

http://www.accessdata.fda.gov/spl/data/0f5196bb-d04d-42f1-865a-

9aea8c7194ea/0f5196bb-d04d-42f1-865a-9aea8c7194ea.xml

[91] Dexbrompheniramine Maleate 2 mg,

http://www.accessdata.fda.gov/spl/data/43dc22bc-37a8-4220-950b-

3c0677ffa794/43dc22bc-37a8-4220-950b-3c0677ffa794.xml

[92] Android SDK, https://developer.android.com/sdk/installing/index.html

[93] GSON, https://code.google.com/p/google-gson/

[94] FDALabel - Full-Text Search of Drug Label Database,

http://www.fda.gov/ScienceResearch/BioinformaticsTools/ucm289739.htm

[95] C. Young, WF Rules and MS BRE - Comparing Performance, 2007.

http://geekswithblogs.net/cyoung/archive/2007/08/12/114597.aspx

[96] Jess, http://www.jessrules.com/jess/index.shtml

[97] Banking Benchmark Performance: Jess vs. Drools, http://illation.com/bankingbenchmark-performance-jess-vs-drools/

## APPENDIX A

## OTC DRUGS USED IN THE APPLICATION

The table below lists the various OTC drugs that our framework currently

supports, all of their active ingredients and the total number of rules written in the

system.

Medicine name	Active Ingredients	Total number	Number of
		of Rules	Facts added
		authored	to working
			memory
<b>Tylenol Sinus</b>	Acetaminophen 325 mg,	13	5
<b>Congestion and Pain</b>	Guaifenesin 200 mg,		
Severe	Phenylephrine HCl 5 mg		
Tylenol ES	Acetaminophen 500 mg		3
Tylenol PM	Acetaminophen 500 mg,	12	4
	Diphenhydramine HCl 25		
	mg		
Robitussin Daytime	Acetaminophen, USP 325	14	5
Cold and Flu	mg,		
	Dextromethorphan HBr,		
	USP 10 mg,		
	Phenylephrine HCl, USP		
	5 mg		
Arthritis Pain Relief	Acetaminophen 650 mg	7	3
Prilosec OTC	Omeprazole magnesium	12 3	
	delayed-release tablet		
	20.6 mg (equivalent to 20		
	mg omeprazole)		
Motrin IB	Ibuprofen 200 mg	18 3	
	(NSAID)		
Dologen	Acetaminophen 650 mg,	15 4	
	Dexbrompheniramine		
	Maleate 2 mg		
All Day Pain Relief	Naproxen sodium 220 mg	16	3
Sudafed	Pseudoephedrine HCl 30	8	3
	mg		

#### APPENDIX B

#### SAFETY RULES FOR OTC DRUG - DOLOGEN

Below are the rules representing safety warnings for OTC drug named Dologen. The rules are divided into three files named Dologen.plsf, Acetaminophen.plsf and Dexbrompheniramine Maleate.plsf.

a) Dologen.plsf

1. (defrule Dologen-alcohol "Alcohol warning"

=>

(add (new Advice (SafetyStatus.CAUTION) "Severe liver damage may occur if you take 3 or more alcoholic drinks every day while using Dologen.")))

2. (defrule Dologen-age-restriction "Restricted age for Dologen administration"

(User {userAge < 12})

=>

(add (new Advice "Unsafe Age: Do not use Dologen in children under 12 years of age.")))

3. (defrule Dologen-healthConditions "Dologen contraindication w/ health conditions"

(User (conditionList ?conditions))

(or (test (?conditions contains "heart disease"))

(test (?conditions contains "high blood pressure"))

(test (?conditions contains "thyroid disease"))

(test (?conditions contains "diabetes")))

=>

(add (new Advice "Ask a doctor before using Dologen if you have heart disease,

high blood pressure, thyroid disease or diabetes.")))

- b) Acetaminophen.plsf
- 1. (defrule Acetaminophen-overdosage "Acetaminophen overdose"

(User (OBJECT ?us))

(test (?us exceedsDrugDoseInWindowHours "Acetaminophen" 4000 24))

=>

(add (new Advice " Unsafe to take this drug as it may cause dangerous overdose of Acetaminophen ")))

2. (defrule Acetaminophen-allergy "Allergic to Acetaminophen"

(User (allergyList ?allergies))

(test (?allergies contains "Acetaminophen"))

=>

(add (new Advice "Unsafe as allergic to Acetaminophen.")))

3. (defrule Acetaminophen-healthConditions

"Acetaminophen contraindication w/ health conditions"

(User (conditionList ?conditions))

(test (?conditions contains "liver disease"))

=>

(add (new Advice "Users with liver disease should not take Acetaminophen.")))

4. (defrule Acetaminophen-bloodThinningMed

"No concomitant use of Acetaminophen with blood-thinning medicines"

(User (prescriptionList ?prescriptions))

(test (?prescriptions contains "Warfarin"))

=>

(add (new Advice "Unsafe to take Acetaminophen with blood-thinning drug Warfarin.")))

c) Dexbrompheniramine Maleate.plsf

1. (defrule Dexbrompheniramine-Maleate-healthConditions

"Dexbrompheniramine Maleate contraindication w/ health conditions"

(User (conditionList ?conditions))

(or (test (?conditions contains "Emphysema"))

(test (?conditions contains "Chronic Bronchitis"))

(test (?conditions contains "Glaucoma")))

=>

(add (new Advice "Users with Emphysema, Chronic Bronchitis or Glaucoma should not take products containing Dexbrompheniramine Maleate.")))

2. (defrule Dexbrompheniramine-Maleate-general "sedatives or tranquilizers"

=>

(add (new Advice (SafetyStatus.CAUTION) "Ask a doctor or pharmacist before use if you are taking sedatives or tranquilizers.")))