Semantic Selection of Healthcare Apps

Hamda Binghubash Almarri
Faculty of Science and Technology
University of Westminster,
London, UK
h.binghubash@my.westminster.ac.uk

Radmila Juric
Faculty of Technology and Maritime Studies
HBV University College, Norway
rju@hbv.no

Bilal Mughal
Faculty of Science and Technology
University of Westminster,
London, UK
b.mughal@my.westminster.ac.uk

Abstract

The paper proposes a computational model, based on SWRL enabled OWL ontologies, in order to create software tools, which will choose the most suitable healthcare Apps for specific requirements. The tool will take into account specific evaluation criteria for the semantic selection of healthcare Apps, which will be based on functionality of Apps, preferences of the Apps’ users and the technology required to run the Apps. We call our selection “semantic” for two reasons: a) we must understand the semantic of the environment where the selection happens before we choose an App and b) we will perform reasoning in order to make a decision on the most suitable healthcare App. Our computational model has been tested in various environments and proved that it can dynamically select Apps according to the semantic found in the Apps evaluation criteria.

1. Introduction

The excessive amount of mobile and wireless software applications, which penetrate every aspect of our lives, has created challenges when choosing the most suitable Apps for our circumstances. Healthcare Apps are not an exception. In principle we want to make sure that the App we wish to download delivers the desired functionality, but it is equally important that it meets personal preferences and operating environments we create as individuals. One approach for evaluating the software in general is to create evaluation criteria with requirements needed for their selection [1], [2], [3], [4], [5], [6], [7]. In this paper we look at the problem of choosing a suitable healthcare App(s) for two reasons:

(i) The number of App downloads is increasing rapidly. The huge number of healthcare Apps that run on various mobile devices is our reality and therefore App selection can be challenging even for the professionals who create/use them.

(ii) Healthcare Apps have become useful and are used in personal/public health and wellbeing and they also supplement traditional, mobile and remote delivery of healthcare [8], [9], [10], [11]. The way forward is to create a universal software tool which chooses the most suitable healthcare App(s) for each patient/healthcare professional based on their specific requirements. Various parties might be interested in it: a) healthcare professionals b) patients who wish to learn/monitor their health conditions through Apps, c) software developers who would learn from the App selection criteria and understand what is needed when building Apps d) device/gadget manufacturers who can improve their current product usability according to a set of requirements defined by the App’s users. It is important to note that, when selecting a healthcare App through the tool, we should not only take into account the App’s functionality. Personal preferences of people who will use Apps and the heterogeneities of operating environments/smart gadgets are equally important.

The novelty of our work is in the automated selection of healthcare Apps based on the semantics of the environment where Apps are needed. We promote decision making upon the most suitable healthcare Apps based on reasoning. One promising way of building such a tool would be to use the Semantic Web Technology stack [12] which produces a computational model based on SWRL (SWRL) [13] enabled Web Ontology Language (OWL) ontologies [14], [15], [16], [17], [18], [19].

In order to illustrate the way the tool would work, we have to focus on a specific domain. We look at Apps for alleviating the management of high blood pressure in individuals by either issuing useful information, tracking their daily activities in order to give advice, warnings and even guidelines for their wellbeing. High blood pressure has been seen as a ‘silent killer’ in our societies because it can lead to heart attacks, strokes, kidney malfunction, cause brain damage, disability and even death [20]. There are many initiatives in healthcare systems which tackle the problem, reduce disabilities caused by this condition and ultimately reduce costs in healthcare systems [21], [22], [23], [24].
The paper is organized as follows. Section 2 gives the background and section 3 describes the problem based on our experiments and investigations. In the Proposal section we create evaluation criteria for selecting healthcare Apps by finding their characteristics and categorizing them. They in turn generate an ontological OWL model which ensures the reasoning upon the semantics stored in it. In section 5 we describe the reasoning, i.e. the decision making on the most suitable healthcare Apps for given circumstances. We comment on related works and give conclusions in the last two sections.

2. The Background

2.1 Mobile Healthcare Apps

In the last decade we have witnessed dramatic changes in mobile/wireless technologies. Mobile healthcare has become a necessity [25], [26], [27], [28]. There have been 75 billion downloads from the Apple store to date [26] and the demand for Apps keeps growing [29], [30]. 80% of doctors use smartphones and tablets at their clinics to support patient care [31] and 93% of physicians use a form of mobile technology [32]. Mobile phones could perform various health related measurements [33], [34], [35], [36], “look” at brain functionality and assess a person’s hearing ability [37]. Therefore healthcare Apps have become a necessity [38]. They are demanded in hospitals because they remove a burden from staff, assist in clinical decision making [8], help doctors in diagnosing diseases [38], improve the interaction between patients and doctors [39] and have an impact on the overall standard of care [40]. Benefits of healthcare Apps from a business perspective are significant and they do not apply only to individuals [41]. By the end of 2015 there will be 500 million smartphone users worldwide using a healthcare App [42] in order to deliver “best practices in a mobile format whenever needed” [43].

2.2 Healthcare Apps for Chronic Diseases

Mobile healthcare has become extremely important in managing chronic diseases. It is based on the fact that it facilitates human interaction and supports mobile collaboration [44]. The need to download Apps on our smartphones was triggered by patients’ inability to keep track of their day to day healthcare management [45]. The regularity of treatment of chronic diseases could be addressed through healthcare Apps, because they can gather and share relevant data between patients and doctors [46].

Apps are useful in the management of chronic diseases due to their user friendly interfaces, the simplicity of their functionalities, and their efficacy in improving patient self-management techniques [43], [47], [38]. One in four adults has chronic health conditions [48] and seven out of ten major causes of death are due to them! Healthcare quality through Apps can meet clinical standards because Apps can assist in decision making tailored for managing chronic conditions. Apps can be crucial in detecting health risks and improving health service, as a part of clinical practices in managing chronic conditions [35], [49], [50].

Healthcare apps for managing high blood pressure are numerous and their role varies. They may recognize whether medical action is required, issue reminders for appointments and notify patients about reports created by Apps. They can raise awareness of health problems and increase the chance of a patient visiting a doctor after detecting unusual symptoms through the Apps.

3. The Problem

3.1 How to Choose a Healthcare App?

Healthcare Apps cover a wide range of health problems: from managing wellbeing, fitness and weight loss to supporting patients with chronic diseases [51]. A large volume of healthcare Apps exists in mobile stores, such as Google, Apple, BB and Windows stores. With the constant evolution of mobile and wireless technologies, plus changes in individual health needs, the decision on the best possible Apps is a complex task. What is the most important aspect to be considered? Is it its functionality? Probably yes. Using Apps without full understanding of their “content” may delay diagnosis for a condition which impacts the patient chance of survival [52]. However, healthcare Apps may support a meaningful interaction between the provider and user of healthcare service [53], but the relation between patients and doctors may deteriorate due to patients’ reliance on healthcare Apps [54].

3.2 Do Healthcare Apps Meet Expectations

Inconsistencies of healthcare Apps functions have been documented. Over 50% of healthcare Apps have been downloaded fewer than 500 times, because users are disappointed with the way the App addresses a particular medical requirement [55]. The readiness of patients to adopt healthcare Apps is also problematic. Elderly patients are likely to benefit
from what Apps provide, but might not feel comfortable when using them. Apps’ interfaces cannot solely resolve the issue of their acceptance by the elderly: functionality will need to be very consistent for them to take Apps’ full benefit. Many Apps do not have the proficiencies required to truly accommodate patients. For example, blood pressure monitors are expedient but they may not provide good instructions on how to operate the device [56].

3.3 Healthcare Apps do not have friendly UI

A good user interface (UI) develops a better overall user experience [57]. A successful healthcare App must have a clear and concise UI: “no matter what you think about your App, if it’s not useable it won’t be used” [58]. In addition, technological growth is inevitable: technology in mobile environments is constantly changing and UI in healthcare Apps will have to address this in order to stay competitive and operational. A friendly UI is very important if Apps are to cater for the elderly. They are sensitive to a device’s screen size, fonts and colors. Elderly people have not grown up with high technology and have special demands regarding its use. A non-user-friendly UI damages the overall quality of the App [59], and poor UI may change the judgement of Apps applicability [60], [61].

3.4 Healthcare Apps may be Costly

There are close to 500 million healthcare Apps users [62]. Do they wish to pay for using them? It’s well-known that users would prefer an App that is free. However, free mobile health apps often require an upgrade after a certain period time [8]. The cost issue goes further because it is often the case that the function within an App which shows readings to doctors might not be free [63]. The criticism of the cost incurred with healthcare Apps focuses on the fact that healthcare delivery may be prohibitively expensive [64]. Healthcare takes up a high percentage of an economy’s expenditure [65], and in countries like the UK, where the NHS provides free treatment [66], the diversion in the management of chronic conditions towards costly healthcare Apps has been criticized: their cost will draw patients away [67].

4. The Proposal

In the previous section we highlighted a few problems with the selection of healthcare Apps which are often overlooked: Apps might not really deliver what their vendors/supplier claim, they can be a costly way of receiving healthcare services and they might not cater for patients who do not embrace new technologies easily. However, we have found attempts to pay explicit attention to usability in Apps, as opposed to their functionality. For example, the Blood Pressure Companion free App in the Apple store has clearly stated the importance of usability by advertising: ‘Easily add and edit blood pressure, heart rate, weight and test date’. The Apps usability was present in other Apple store Apps, such as Blood Pressure + Pulse Grapher Lite and BPMon - Blood Pressure Monitor and also present in Apps on Android platforms (e.g. My Heart). In [68] we find user compliance, intermittent availability of connectivity, reliability of transmission, security, and interoperability of different platforms in Apps specification. Finally feedback in healthcare is a reinforcing factor that ultimately encourages individuals to adopt a more positive lifestyle, as demonstrated in windows App [69]. However not many Apps offer feedback.

To summarize, our proposal should

(a) specify a minimum list of selection criteria for healthcare Apps which is generated from our individual inspections of available healthcare Apps aimed at the management of high blood pressure as a chronic disease and

(b) illustrate a computational model which takes the categorized selection criteria from (a) into account and enables reasoning upon the available data in order to ensure that we can choose the most suitable healthcare Apps for given requirements.

The proposal from (a) is given in section 4.1 and from (b) in section 4.2.

4.1 Criteria for Selecting a Healthcare App

4.1.1 Finding healthcare Apps’ characteristics.
The selection criteria, as noted in (a) above, have to be based on identifying characteristics of Apps which are aimed at the management of high blood pressure.

We used data available on (I) UK NHS websites which address problems for patients who live with this chronic condition and found core characteristics that such Apps should have and (II) data available on App’s URLs in order to understand how commercial Apps are marketed. For example, it was obvious that these Apps should have systolic and diastolic blood pressure reading, followed by the calculation of mean arterial pressure, pulse measurement and possibility of entering body position at pulse measurement (not so common but essential for some people). Consequently body measurement values such as height and weight have medical significance and we
should make provision for assessing if Apps have such or similar characteristics. We have found that some Apps offer food and activity logs, input of gender, input of the type of meal consumed, and feedback and printing of the collected data. Minimum and maximum blood pressure readings are considered important, the tracking of the level of calories in a patient’s diet beneficial, showing the exercise time useful, a reminder setting, as well as giving an option to input personalized notes welcome. Apps also may allow users to take a snap shot of their blood pressure readings, add, edit or delete data inputted, specify water consumption, calculations of weekly, monthly and daily blood pressure readings.

Table 1 gives our initial list of characteristics of Apps used for the management of high blood pressure. They are listed in the order they were discovered through (I) and (II) above.

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Body Functions</th>
<th>Activities</th>
<th>Calculations</th>
<th>IT Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track systolic and diastolic</td>
<td>body height</td>
<td>Food and activity log</td>
<td>Calculate Mean Arterial Pressure (BP)</td>
<td>Reminders</td>
</tr>
<tr>
<td>calculate mean arterial pressure</td>
<td>reminders</td>
<td>show MAX, MIN and AVG number of BP readings</td>
<td>Show MAX, MIN and AVG number of BP readings</td>
<td>Report progress card</td>
</tr>
<tr>
<td>supports printing data</td>
<td>gender</td>
<td>exercise time</td>
<td>Exercise time</td>
<td>Save snapshots</td>
</tr>
<tr>
<td>edit blood pressure</td>
<td>save snapshots</td>
<td>add heart rate</td>
<td>add heart rate</td>
<td></td>
</tr>
<tr>
<td>edit heart rate</td>
<td>add blood pressure</td>
<td>add weight</td>
<td>add weight</td>
<td></td>
</tr>
<tr>
<td>edit weight</td>
<td>add weight</td>
<td>water consumption</td>
<td>test date</td>
<td>Easy editing or deleting any saved record</td>
</tr>
<tr>
<td>pulse measurement</td>
<td>test date</td>
<td>Meal type</td>
<td>Calcuating averages daily, weekly, monthly BP</td>
<td>Supports printing data</td>
</tr>
<tr>
<td>Track systolic and diastolic</td>
<td>calorie tracker</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Blood Pressure</td>
<td>Glucose</td>
<td></td>
<td>Email function</td>
<td></td>
</tr>
<tr>
<td>Add Heart rate</td>
<td>Exercise time</td>
<td>Water consumption</td>
<td>Personalized notes</td>
<td></td>
</tr>
<tr>
<td>Test Date</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit Blood Pressure</td>
<td></td>
<td></td>
<td></td>
<td>Back up/Capability</td>
</tr>
<tr>
<td>Edit Heart rate</td>
<td></td>
<td></td>
<td></td>
<td>Feedback</td>
</tr>
<tr>
<td>Edit Weight</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support lbs and kgs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>body position at measurement</td>
<td></td>
<td></td>
<td></td>
<td>email function</td>
</tr>
</tbody>
</table>

Table 1 Characteristics of Healthcare Apps

4.1.2 Categorizing characteristics of Healthcare APPs

Characteristics of the Apps given in Table 1 have to be characterized for two reasons:
(A) If we wish to use these characteristics as an Apps selection criterion in the automated software tool, based on computations with SWRL enabled OWL ontologies, then we need to place them into a taxonomy which semantically groups these characteristics;
(B) If we wish to extend/amend the list of characteristics we need a structural format which could help in highlighting the deficiencies of the selection criteria/characteristics of Apps.

Therefore, for the purpose of testing our selection criteria and the reasoning with SWRL, we intuitively grouped them into four segments: Body Functions, Activities, Calculations and IT functions. These groupings can be changed in future. We would like to emphasize that the diversity of these groups of characteristics could motivate App developers to look at the range of issues which may determine the suitability of their product in real life.

Table 2 shows a categorized list of Apps characteristics which follows these four groups.

HEALTHCARE APPS CHARACTERISTICS

<table>
<thead>
<tr>
<th>Body Functions</th>
<th>Activities</th>
<th>Calculations</th>
<th>IT Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body height</td>
<td>Food and activity log</td>
<td>Calculate Mean Arterial Pressure (BP)</td>
<td>Reminders</td>
</tr>
<tr>
<td>show MAX, MIN and AVG number of BP readings</td>
<td>Show MAX, MIN and AVG number of BP readings</td>
<td>Report progress card</td>
<td></td>
</tr>
<tr>
<td>Pulse measurement</td>
<td>Meal type</td>
<td>Calculating averages daily, weekly, monthly BP</td>
<td>Passcode function</td>
</tr>
<tr>
<td>Track systolic and diastolic</td>
<td>calorie tracker</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Blood Pressure</td>
<td>Glucose</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Heart rate</td>
<td>Exercise time</td>
<td>Water consumption</td>
<td></td>
</tr>
<tr>
<td>Test Date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit Blood Pressure</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit Heart rate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit Weight</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support lbs and kgs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Categorized list of App characteristics

4.2. Computational Model based on the Selection Criteria

The computational model with SWRL enabled OWL ontologies, which will become a core of the tool for the semantic selection of healthcare Apps, must include the semantics of categorized characteristics of Apps from Table 2 and deploy it in the OWL model. We need to show OWL classes and subclasses, their individuals and constraints. The model should also show through the reasoning upon OWL concepts and results which healthcare App is the most suitable for given requirements.

4.2.1 Ontological Modeling

In this study we have decided to use the OWL model, tested in in different domains, but used for the same purpose: filtering various information (SOURCES) according to given requirements
(USER_PREFERENCES) [70], [71], and [72]. The model is given in Figure 1. The selection and reasoning is based on the description of the class SOURCES through their characteristics, which are stored in CS\(_i\) classes. CS\(_i\) classes have relationship with SOURCES / USER_PREFERENCES classes which is defined by has_CS\(_i\) constraint (object property). We can have as many CS\(_i\) classes as needed to describe characteristics of SOURCES / USER_PREFERENCES classes. The semantic overlapping is shown through constraints has_CS\(_i\); we impose has_CS\(_i\) on both SOURCES and USER_PREFERENCES classes which will allow us to describe individuals of SOURCES and USER_PREFERENCES classes similarly. We can achieve the semantic matching of individuals of SOURCES and USER_PREFERENCES through SWRL rules. The results are exact matches: we list individuals of SOURCES which match individuals of USER_PREFERENCES.

**Figure 1. The Generic OWL Model for the Selection of SOURCES**

### 4.2.2. The Proposed OWL model

Figures 2 & 3 show the OWL ontological model generated from Figure 1. Figure 2 is self-explanatory. However, in order to mirror the semantics from Figure 1 we need to:

1. Describe classes BP-APPS-SOURCES and PATIENTS-PREFERENCES, by defining their characteristics, i.e. we must find CS\(_i\) classes which will be added to the model from Figure 2.
2. Define a set of object properties which will allow us to link individuals from BP-APPS-SOURCES/ PATIENTS-PREFERENCES classes to a set of characteristics classes CS\(_i\).
Our set of CS\textsubscript{i} classes are taken from Table 2: column headings are CS\textsubscript{i} classes, and values in each column of table 2 can become subhierarchies of the CS\textsubscript{i} classes, which is shown in Figure 3. Therefore Figure 3 mirrors Figure 1 fully.

Figure 4 shows constraints upon OWL model as a list of object properties has\_CS\textsubscript{i}. Each of them enriches the description of individuals from BP-APPS-SOURCES/PATIENTS-PREFERENCES classes. Note: we have an extra property named has-OP-System because we singled out one special characteristic which was not a column name in Table 2: we wanted to have an operating system where an App is running explicitly available at the base class level. Furthermore, subhierarchies of CS\textsubscript{i} classes contain individuals which bear the same name as subclass hierarchies (see Table 2). If we wish in future to make distinctive names for individuals of CS\textsubscript{i} class subhierarchies the model and reasoning will remain the same.

In Table 3 object properties from Fig. 4 connect individuals from base classes in Fig. 2. Class BP-APPS-SOURCES is related to class ACTIVITIES through object property has\_Activities. BP-APPS-SOURCES is a domain (D) and ACTIVITIES is a range (R) class for object property (has\_Activities). A detailed picture of relationships through object properties between individuals of domain and range classes extends what is in Table 3. Table 3 is an illustration of how properties work in our model.

If we wish to show which exact individuals are stored in BP-APPS-SOURCES class then Figure 5 gives a screen shot of its content. We can see that individuals of the BP-APPS-SOURCES class create a pool of Apps which will be selected through the reasoning according to given requirements.

5 The Reasoning

The reasoning process has to show two things:

- We must say exactly why the reasoning is needed and what it will achieve
- We have to show which classes from the OWL model are involved in reasoning and which SWRL will be used.

<table>
<thead>
<tr>
<th>D-OB</th>
<th>R-Class IND</th>
<th>R-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-IND</td>
<td>OB</td>
<td>ACTIVITIES</td>
</tr>
<tr>
<td>BP-Tracker has-Activities</td>
<td>Meal-Type</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-Activities</td>
<td>Exercise-Time</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-Activities</td>
<td>Glucose</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-Activities</td>
<td>Calorie-Tracker</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-Body-functions</td>
<td>Add/edit-Weight</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-Body-functions</td>
<td>Support-Lbs-And-Kgs</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-Body-functions</td>
<td>Add/Edit-BP-Heat-Rate</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-BP-Heat-Rate</td>
<td>Add/Edit-Gender</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-BP-Heat-Rate</td>
<td>Pulse-Measurement</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-BP-Heat-Rate</td>
<td>Add/Edit-BP-Test-Date</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-BP-Heat-Rate</td>
<td>Add/Edit-BP</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-BP-Heat-Rate</td>
<td>Add/edit-Body-Temp</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-Calculations</td>
<td>Calculate-Mean-Arterial-Pressure-Bp</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-Calculations</td>
<td>Show-Max-Min-Avg-No-Of-Bp-Readings</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-IT-Functions</td>
<td>Reminders</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-IT-Functions</td>
<td>Feedback</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-IT-Functions</td>
<td>Report-progress-card</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-IT-Functions</td>
<td>Passcode-function</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-IT-Functions</td>
<td>Personalized-notes</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-IT-Functions</td>
<td>Easy-editing-deleting-saved-record</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-IT-Functions</td>
<td>Save-snapshots</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-IT-Functions</td>
<td>Email-function</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-IT-Functions</td>
<td>Supports-printing-data</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-OP-Systems</td>
<td>BlackBerry</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-OP-Systems</td>
<td>Windows</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-OP-Systems</td>
<td>OIV</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-OP-Systems</td>
<td>IOS</td>
<td></td>
</tr>
<tr>
<td>BP-Tracker has-OP-Systems</td>
<td>Andriod</td>
<td></td>
</tr>
</tbody>
</table>

In the generic model from Figure 1 we showed the reasoning process in the right part of the figure. Therefore we will have three classes involved in our reasoning: BP-APPS-SOURCES and PATIENTS-PREFERENCES classes as base classes and class RECOMMENDED-APPS where we will store the result of reasoning. Individuals from the BP-APPS-SOURCES class, given in Figure 5, will be selected and moved into the RECOMMENDED-APPS class if they satisfy specific requirements. We illustrate these requirements in the scenario below.
Mary suffers from high blood pressure and would like to download an app on her iPhone to help her to manage/monitor her daily activities, water consumption, and keep track of calorie intake. Mary also wishes to be able to record her body functions such as add/edit her BP count, add/edit personal data such as body weight, eating habits and keep a record of her frequency of BP test, and dates. She prefers to download an app which provides basic email functionality and gets her calculations of her daily min. max and avg BP.

The text above immediately generates a set of object properties which connect PATIENT-PREFERENCE class (underlined words) and characteristics classes from Table 2. Figures 6 shows how BP-Monitor-Family-Lite App is described in the model. The reader should note that each line in Figure 6 gives the exact name of object properties which connect BP-Monitor-Family-Lite individuals with the selection of characteristic classes CS_i.

Figure 7 shows an instance of the asserted semantics of Mary’s preferences in the proposed ontological model. Therefore the text above instantly creates object properties which describe Mary’s requirements. These properties will create ontological matching with the characteristics of all Apps stored as individuals of the BP-APPS-SOURCE class from Figure 5.

The SWRL rule in Fig. 8, examine all individuals listed in Fig. 5 and though the object properties singles out BP-Monitor-Family-Lite as an App which is suitable for Mary’s requirements. This result of reasoning: BP-Monitor-Family-Lite is now an individual of the RECOMMENDED-APPS class, as shown in Fig. 9. It is important to note that when we collect all individuals of the BP-APPS-SOURCE class, we describe them though their characteristics, and therefore the BP-Monitor-Family-Lite individual has been described through object properties given in Fig. 6.
6. Related Work

We could not find published related work. There are studies which look at, analyze and evaluate Apps available on the market. In [73] the authors grouped a total of 400 Android Apps to analyze interface layouts. The study from [74] looked at Apple’s categorization of healthcare Apps within the App store, but their categorizations has no taxonomy of functionalities these Apps offer. Healthcare Apps from Apple and Android were investigated and grouped into 11 categories n [75]. The grouping may help in searching for Apps within a particular category instead of going through all available Apps.

7. Conclusion

The idea of selecting the most suitable healthcare Apps for given requirements has been promoted and utilized in this paper by proposing a computational model based on SWRL enabled OWL ontologies which have to have the characteristics of Apps and requirements for them stored within the OWL model. The ontological matching between the two: Apps’ characteristics and given requirements is used in reasoning in order to determine exactly which Apps suit which requirements. The proposed computational model is close to a full scale software tool, which can be implemented as a Java or Android based application in distributed environments using our previous experiences of integrating SWRL reasoning in Java applications [76, 77]. In practice, user may make decisions on their requirements either by clicking on the suggested drop-down menus or by choosing keywords which relate to possible requirements the tool may cater for [78]. Ad-hoc changes in user’s decisions are addresses by re-running the application: the results of previous “reasoning” will be deleted. “Illogical” user behavior may end up with “no-results” after the reasoning. This can trigger a dual action: App developers learn that they have NOT developed some characteristics of their Apps which are needed by users and users should focus on drop-down menus/keywords in order to benefits from tool’s semantic selection of Apps.

Our proposal is flexible: SCi classes might be dynamically inferred through the tool to create Fig. 3. This will happen in future if categorized characteristics of Apps from Table 2 change. Object properties may also be inferred as a result of collecting patient preferences and consequently, we can achieve a high level of automation with this model. It is important to note that there will be a certain number of OWL concepts which will always be in the OWL ontology and there will be concepts which will have to be removed from the ontology. RECOMMENDED-APPS individual(s) should be removed from the class once we learn which App is recommended for a particular patient. Changes in the Apps market/mobile technologies must allow for newcomers in our OWL model and it is not very prudent to keep results of the reasoning upon this model as “knowledge” assuming that it will always be valid. Our experiments with Apps on high blood pressure proved the concepts of the proposal, but it can be used for any another type of App.

8. References

th nhs

http://mobihealthnews.com/41727/englands-nhs-launches-


Process Science, SDPS 2013 Proceedings of the 18th Int. Conf. on System Design and Decision Making in Traffic For Individual Driver In

States, http://www.brookings.edu/events/


Pharmaceutical Medicine, 29(1), 2015, pp. 17-25.


[58] Leslie, P., Up at 5am: The 5am Solutions Blog, http://info.5amsolutions.com/blog/bid/156167/6-Keys-to-a-User-Friendly-Mobile-Health-App
[74] Hussain, I., Apple Tries to Help Doctors Categorize the Medical Apps Section, but Falls Short, http://www.imedicalapps.com/author/iltifat/page/14/

3287