MISS: Medicine Information Support System in the Smart Home Environment

José M. Reyes Álamo, Johnny Wong, Ryan Babbitt, and Carl Chang

Department of Computer Science
Iowa State University
Ames, Iowa 50011
{jmreyes,wong,rbabbitt,chang}@cs.iastate.edu

Abstract. The Smart Home uses different technology to facilitate the lives of the resident and is especially useful for assisting the elderly and persons with special needs. One area where this population would benefit is managing their prescribed medications. This paper presents the Medicine Information Support System (MISS) which integrates the patient's information to assist with the prescriptions management. The system checks for conflicting medicines, health conditions and food items. The data generated is used to feed other subsystems in the Smart Home such as the reminder and medicine inventory. A formal model is introduced for conflicts checking. The three main entities: doctor, pharmacy and Smart Home use this model to detect their particular set of conflicts which ensures that conflicts involving the entire context will eventually be detected. The design uses this model as its basis for conflict checking. The prototyped implementation of the entire system is based on Java.

Keywords: Smart Home, doctor, medicine, pharmacy, formal model, conflict checking, RFID, OSGi.

1 Introduction

The Smart Home is a house that integrates different technologies for facilitating the execution of daily tasks. Sensors and actuators play a major role in assisting in the automation of these tasks. Smart Homes, with modern technology designed especially for the elderly and persons with special needs, have been a research subject in the last few years. One main motivation for this research is that the baby boomer generation is reaching the retirement age and the need for assistance increases as they grow old.

One of the areas in which the elderly and persons with special needs would need assistance from the Smart Home is in their medicine intake management. Keeping up-to-date with the prescriptions can be challenging due to complicated medicine names, several simultaneous medications, similar instructions for medication intake for different medicines, and being aware of expiration dates and detecting conflicts.

We propose MISS: Medicine Information Support System in the Smart Home to address this issue. A step-by-step analysis of the process for getting prescriptions today is presented. This analysis helped identifying important requirements for our

S. Helal, S. Mitra, and J. Wong (Eds.): ICOST 2008, LNCS 5120, pp. 185–199, 2008.

system. As a result we come up with a formal model which is used for detecting errors and conflicts among sets. These sets are the medicines, food items and patient's health conditions and the system checks for conflicts among those. This system generates data which can be used by other subsystem in the Smart Home. The reminders system which tells the patient when to take the medicines is one of these subsystems that benefits from MISS. Also MISS data is useful for preparing and updating a personalized calendar and providing individual assistance for each user at the time of taking the medicine.

There have been previous efforts for helping individuals with the management of their prescriptions. Some related work includes the Magic Medicine Cabinet (MMC) which is presented in [1]. In that project the author mentioned that today's smart devices are designed to perform the device tasks plus connecting to the Internet. This converts the appliance to a device similar to a personal computer which also can go online. MMC is equipped with a facial recognition software, RFID smart labels, vital signs monitor and voice synthesis. The MMC assists the residents of the house by giving personalized reminders, detecting when a resident take the wrong medicine, and measure some vital signs. The MMC is a great idea but their product is not designed particularly for the needs of the elderly population. They also do not give details of how it interacts with the patient's pharmacy, doctors and health care providers even though the MMC claims it can. Our paper bridges this gap by describing a system in which the Smart Home interacts with the patient's doctor, pharmacy and health care provider. Such a system will be useful for checking conflicts and errors in the process of dispatching medicines, will facilitate giving reminders, and will increase compliance with medication intake [7].

The Smart Medicine Cabinet [2, 3] and the Smart Box [4, 5, 6] extends the Magic Medicine Cabinet by using passive RFID technology and Bluetooth, to synchronize the state of the MMC using a cellular phone. The cellular phone contains information to be used to give reminders and to know the state of the medicine cabinet and its content. They assumed that the medicine containers have RFID tags and the Smart Medicine Cabinet (SMC) can be automatically updated. When the cell phone is within the SMC range, a synchronization phase takes place keeping user intervention to a minimum. Nevertheless this still requires the user to carry the cell phone to the synchronization area as well as to carry it to the pharmacy. Our system presents a simpler synchronization process from the patient's point of view that will not require any intervention from the patient. Also there will be no need of carrying any device such as a cell phone giving our system another advantage.

Technology available for automatic dispensing of prescribed pills can be found in [10, 11]. All these products have some common features. In these machines either the resident of the Smart Home or a caregiver has to load the automatic dispenser with the medications, enter the times that the medicine should be taken, remove the medicines after the system reminds the patient for taking them and repeat these actions for other prescriptions. The disadvantage is that these machines require a lot of manual action.

All these products has some outstanding features that facilitate the task of taking medicines [10, 11], ensuring that the right medicine is taken at the proper time [4, 6] and giving reminders to the patient [8, 9]. Also these products facilitate to some extent the detection of errors during the process such as when the medicines are not taken. But these products by themselves do not address the need of managing medicine information as they still require a lot of manual input from the patient. In [13] a system which uses this technology for helping patients with dementia is presented. Our system is similar in which it uses available technology to help patients with their prescription intake. Our system is different as it involves the doctor, pharmacy and health care providers at early stages. Also we provide a formal model of the system to ensure it is safe, secure and correctly detects the conflicts. Also MISS presents a system in which the patient does not need to enter any data manually and can be integrated with existing reminder systems such as outlined in [13]. The following sections will describe more details of our system MISS. The rest of the paper is organized as follows: Section 2 lists the system requirements. Section 3 describes MISS details and design. Section 4 introduces a formal model of the system. Section 5 explains the conflicts checking at each subsystem. Section 6 shows an instantiation of the model. Section 7 presents the prototyped implementation. Section 8 concludes the paper and presents future work.

2 Current Technology and System Requirements

Describing the process of a person who goes to the doctor and receives a prescription will help to identify important system requirements. This process can be broken down into the following steps:

1. The person visits the doctor.
2. The doctor prescribed some medicines.
3. The patient goes to the pharmacy and gets the prescribed medicines.
4. The patient goes home and intakes the medicines.

Medicines intake involve the following steps:

4.1 Wait for the next dosage time
4.2 Locate the medicine container
4.3 Open the container
4.4 Extract the appropriate amount of medicine
4.5 Intake the medicine
4.6 Close the prescription container
4.7 Return the container to the medicine cabinet

Several of these steps have been automated with existing products. For step 4.1 and 4.2 a Smart Box or Cabinet can help with the reminders and location of the medicines [1, 2, 4, 6]. Using automatic pill dispensers [10, 11] can help in opening the medicines and extracting the right amount as indicated in steps 4.3 and 4.4. These are some examples on how current technology can be used for the purpose of medicine intake but there is still room for improvement. One of the steps which can be further improved is 4.1 where the patient “waits” for the next dosage time. The elderly and persons with special needs might forget when the next dosage is [13]. A system which takes care of reminding them the time of the next dosage will help to increase compliance with medicine intake [7]. Another problem facing this population is to locate the medicine as required in step 4.2. It is possible that they do not remember where the medicine containers were placed last time. Therefore the patient will benefit from a system that will help locating the medicine containers. The automation of a reminder system will require input of the prescription's specific information to be able to give the proper reminder at appropriate time. To enable the location of medicines, a unique identifier for the prescription containers is needed. This will facilitate tracking the object within the Smart Home and distinguishing it from similar products. An efficient mechanism to detect and locate the medicine container is required.

Entering the medicine information manually by the resident or a health care provider for these systems is not feasible. Human errors and typos can occur. Therefore the details about the prescription such as the name, dosage, conflicting medicines, conflicting food, patient conditions and other warnings should be entered by an expert and automatically transferred into the Smart Home system. This way intervention from the resident is minimal so this feature is very important. All this automation will require a reliable system which will accurately compute the existence of conflicts among the prescribed medicines, the food items available at the Smart Home, and the patient health conditions. This data must be accurate as it is expected to be used as input for other subsystems in the Smart Home. The design and the prototyped implementation of the model show how these tasks can be accomplished by our system requiring minimum intervention from the patient. The next section shows the MISS system and design.

3 MISS System and Design

This section has a high level description of the MISS system, it subsystems and their interactions with each other for managing the medicines and detecting conflicts. At a very high level the system should do the following: The patient visits the doctor and gets a prescription. The prescription details are inputted into the system at the doctor's office. The system will check for conflicting medicines and heath conditions based on the patient's record that the doctor has. The patient will indicate from which pharmacy the prescription will be picked up. The doctor then will make the prescription's data available to that particular pharmacy. The pharmacy prepares the prescription based on the doctor's prescription. The pharmacy will double check for conflicting medicines and health conditions based on the patient's record of the pharmacy. The patient picks up the medicines. The patient goes to the Smart Home and in a very convenient way, by scanning the RFID-enabled prescription containers into a RFID-reader, will indicate to the Smart Home system the presence of the new medicine. The Smart Home will update its medicine inventory database accordingly and makes a final check for conflicts among medicines, health conditions and food. MISS consists of three main subsystems: The Doctor Subsystem, the Pharmacy Subsystem and the Smart Home Subsystem. These subsystems are operated by four main actors which are: the doctor, the pharmacy, the patient and the Smart Home. A trusted third party medicine's database that defines the conflicts is also used. A diagram of the system is shown in Figure 1. The next subsections describe each subsystem design more detailed.

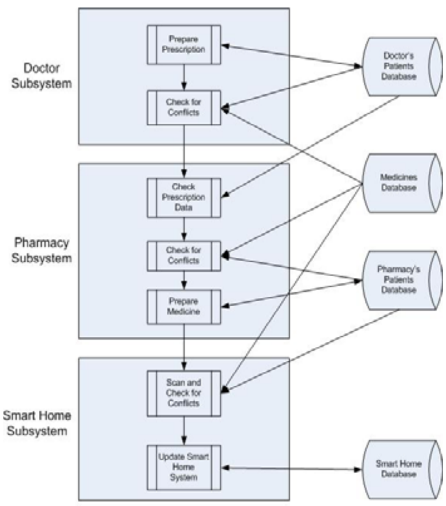

Fig. 1. Medical Information System Diagram

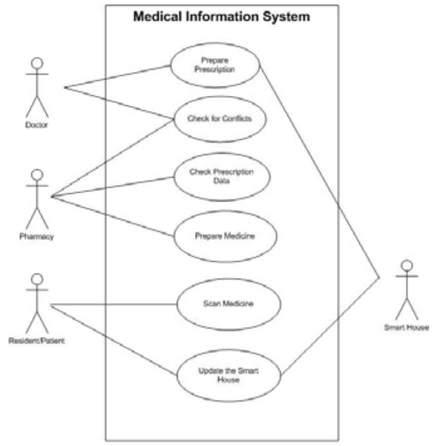

Fig. 2. Use Cases and Actors

3.1 Doctor's Subsystem

The doctor subsystem is where the process starts. During the visit to the doctor, he checks the patient and prescribes some medicine. During the consultation the prescription details such as the name of the medicine, dosage, etc. are inputted into the system. Our system checks for conflicts and also facilitates the communication between the doctor's office and the patient's preferred pharmacy. For doing this the options are to have the doctor's office directly communicate with the patient's preferred pharmacy and send the prescription data or making the prescription data available for a pharmacy that the patient will choose later.

Consider the first case when the patient chooses a preferred pharmacy to pick up the medicines. The person interacting with the system at the doctor's office will use a unique ID such as an assigned patient number, to access the patient's information. This information is stored at a local patient's database, available only to the doctor. The information extracted from the database contains data such as previous prescriptions and health conditions. To ensure that the prescription would not have any adverse side effect on the patient the information extracted from the database will be checked against the data of the new prescription. After carefully checking for conflicts with medicines and health conditions, if no conflict is found the prescription data will be sent to the patient's preferred pharmacy through a secure channel. The doctor's office will also issue prescription document customized for that particular patient and prescription. This prescription will be in the form of printed-RFID tag which will be used later by the pharmacy module.

In the second case in which the patient will decide later from which pharmacy the prescription will be picked up, the process is similar. At the doctor's office the prescription information will be entered into the system, the patient's data will be accessed and check for conflicting medicines and health conditions as in the previous case. But here the doctor's office will not be sending the data to any pharmacy.

Instead a printed-RFID prescription will be issued. This will allow the pharmacy's system to download the prescription data from the doctor's office later on. The main difference among the two approaches is when the data arrives to the pharmacy. In the first case it arrives immediately in the second case when the patient gets into the pharmacy. Now let us consider how the pharmacy subsystem uses this data.

3.2 The Pharmacy Subsystem

Based on the doctor's subsystem operation we assume the following two starting scenarios for the pharmacy subsystem: There is a chosen pharmacy which receives the prescription's data from the doctor's office or the patient will choose a pharmacy later and bring the printed-RFID prescription.

Consider the first scenario in which the patient chose a preferred pharmacy at the doctor's office. The pharmacy will receive the prescription's data with the necessary details when the patient is still at the doctor's office. The pharmacist can start preparing the prescription immediately using this information. The pharmacy will issue the prescription in special containers. These containers will look like regular ones with the difference that they will be equipped with RFID tags. These RFID tags will allow the system to uniquely identify the particular medicine with all its related data. This tag-ID and the related information will be stored in a database that will be used later by the Smart Home module.

Before assigning the RFID tags for the container, the pharmacy system should have available a history of the patient and previous prescriptions dispatched from that pharmacy. The system will check that the particular RFID tag has not been assigned maintaining uniqueness. After assigning unique IDs to each prescription, the system is ready to prepare the data that will be used for updating the Smart Home subsystem. This data consists of two important pieces: the patient independent and the patient dependent information. The patient independent information contains the description of the medicine, possible side effects, different conflicts and recommendations. The patient dependent information is the historical data that the pharmacy has about the patient. This information will be checked in a similar way as it was checked in the doctor's module. The system will be looking for possible conflicts with other medications, and health conditions to make sure it is safe to take that particular medicine. This double check is necessary as the doctor might prescribe a medicine which creates a conflict with a medicine previously picked up at that pharmacy.

At this point the pharmacy is ready to receive the patient. When the patient arrives he shows the printed-RFID prescription the same way they are used to do it now. The difference is that the prescription ready or in process as the patient chose the preferred pharmacy when still at the doctor's office. The printed-RFID prescription is scanned in a RFID reader. At this point the system will compare the data in the RFID tag with the data received from the doctor's subsystem. If no incongruence is found, the medicine is dispatched. Otherwise, the pharmacist is alerted and contacting the doctor's office is recommended. One advantage of our system is that it reduces the waiting time at the pharmacy as pharmacists can start preparing the prescription when the patient is still at the doctor's office. This is an excellent feature especially for the elderly population that might need their medicines as soon as possible or want to avoid long waits or several trips to the pharmacy. Another benefit of our system is the double layer of security checking for conflicting medications and health conditions.

In case that the patient did not choose a preferred pharmacy the waiting time will increase but the process will be almost identical to the one described previously. Instead of having the prescription ready or in process, the patient starts the process of getting the prescriptions when arriving into the pharmacy. At the pharmacy counter the pharmacist will receive the printed-RFID prescription and at that moment the pharmacy system will download the prescription data from the doctor's office and perform all the safety checks for conflicting medicines and health conditions described in the previous paragraphs. At this point whether the patient pre-selected the pharmacy or not, the medicines should be ready and the patient can go home. The process for updating the system in the Smart Home will be very simple from the patient's point of view as the underlying system will take care of all the details.

3.3 The Smart Home Subsystem

This is the moment in which the patient finally arrives to the Smart Home and updates the subsystem with the new prescription's data. Either the patient or a caregiver will be in charge of updating the Smart Home system by scanning each prescription container with an RFID reader. After scanning the prescriptions the medicines can be placed in Smart Medicine Cabinet [2, 4] or loaded into an automatic medicine dispenser [10, 11] or a combination of both technologies for storing the medicines.

At this point the Smart Home system will read the RFID-tags of the prescription container. These tags will contain information indicating from which pharmacy the patient picked up the medicines. The Smart Home will have a secure communication link to the pharmacy. A query will be issued to the pharmacy to retrieve the prescription details and download this data to the Smart Home subsystem, similar to the process of downloading the data from the doctor to the pharmacy. A final safety check for conflicting medicines, health conditions and food items will take place. This check is necessary as the patient might be picking up the medicines from different pharmacies, prescribed by different doctor. The check for conflicting food will be performed at the Smart Home level as it is the subsystem with the database of available food items.

If a conflict is detected then a caregiver will be informed. If no conflict is detected then with all this information, the Smart Home System will update its medicine inventory and provide data to other subsystems such as the reminders and personalized calendar. All these tasks that MISS perform will definitely help the elderly and persons with special needs that have a hard time checking all these safety issues by themselves.

The more important use cases and actors for the MISS system are shown in *Figure 2*. The system will have four main actors: the doctor, the pharmacy, the patient and the Smart Home. From the figure we can see that the doctor actor is in charge of starting the system by preparing the prescription and make it available to the pharmacy. The doctor actor will also be the first one to check for conflicting medications and health conditions. The pharmacy module then will use the information provided by the doctor's office to prepare the appropriate prescription. It will also check for conflicts and will make sure to validate that the data received from the doctor's office is correct. The patient actor will scan the RFID enabled prescription containers to feed the system with the information necessary to obtain the prescription details from the pharmacy's database. The Smart Home actor will then download the prescription's data from the pharmacy and will use it to update it medicine inventory and support the reminder, calendar, notification and location and tracking subsystems. To ensure the accuracy and correctness of the conflict detection a formal model is proposed in the next section.

4 Proposed Model

The medicine management system should be accurate, reliable and provide safety by detecting and informing conflicts. A conflict occurs when a medicine should not be taken together with another medicine, if the patient has certain health condition that the medicine could aggravate or the medicine interacts with certain food items. We are assuming that a trusted third party defines the conflicts and these definitions are publicly available for the model and the system to use them. To check that these kind of conflict do not occur we present the following model whose main components are the set of medicines *M*, the set of food items *F* and the set of medical conditions *C*. The set of food items *F* needs to be considered as some food should be avoided when taking certain medicine. The set of medical conditions *C* needs to be considered as some medicines cannot be taken if the patient has certain conditions.

Now we define several functions that will act over the set of medicines and return useful information. The first function is *conflicting_medicines: M→P(M)*, where *P(M)* is the powerset of the set *M,* which determines the set of conflicting medicines. The next function is *conflicting_food: M→P(F)* where *P(F)* is the powerset of the set *F,* which will return the set of conflicting food items. The other function is *conflicting_conditions: M→P(C),* where *P(C)* is the powerset of the set *C,* which will return the set of conflicting health conditions. Given these sets and function we define the Medicine System Model as follows:

Definition
A Medicine System Model *S* consists of the following sets:

- M, the set of medicines
- C, the set of medical conditions
- F, the set of food items
- D, the doctors, hospitals or clinics the patient visits
- P, the set of pharmacies at which the patient gets prescriptions
- H, the patient's Smart Home

With the following functions:

- conflicting_medicines: M→P(M)
- conflicting_conditions: M→ P(C)
- conflicting_food: M→ P(F)

We want this information to be useful for a particular patient *p*. Each patient will be represented by a tuple.

p = (id, Mp, Cp, Fp, CMp, CFp, CCp). The *id* entry will uniquely identify the patient. Let $Mp \subseteq M$, represents the subset of medicines prescribed to patient *p*. Let $Cp \subseteq C$ represent the health conditions that patient *p* has been diagnosed. Let $Fp \subseteq F$ represent the food items that patient *p* has available. Let *CMp* represent the subset of medicines that are currently in conflict with the medicines prescribed to patient *p* and therefore should not be prescribed to that patient. *CMp* can be computed as $\bigcup_{m \in Mp} conflicting\_medicines(m)$. Let *CCp* represents the set of medical conditions that the patient should not have in order to take the medicine safely. *CCp* can be computed as $\bigcup_{m \in Mp} conflicting\_conditions(m)$. Let *CFp* represents the set of food items that the patient should avoid while taking the medicines prescribed to him. *CFp* can be computed as $\bigcup_{m \in Mp} conflicting\_food(m)$. This will help to detect when a patient is diagnosed with a health condition and is taking a medicine which is in conflict, or when a medicine is prescribed which is in conflict with an existing health condition. This way the doctor or caregiver can make better decisions.

Now we describe how to construct these sets that complete the patient information. Given a new prescribed medicine *m*, to a patient *p*, the medicine has some data related to it like the unique drug id, active ingredients, milligrams, and so on. The medicine data can be obtained from the Food and Drug Administration (FDA) [14], or from the Physician's Desk Reference (PDR) [15]. These entities are trusted third parties who define the conflicts among medicines, food and health conditions. These definitions of conflicts will be used in our model and system. Using the data of medicine *m,* we compute $CM = conflicting\_medicines(m)$, which returns the set of conflicting medicines with medicine *m*. We check if there is a conflict by examining if $CM \cap Mp = \emptyset$ and $m \cap CMp = \emptyset$. If both conditions are true then we take $CMp = CMp \cup CM$, to update the set of conflicting medicines for patient *p*. We do a similar processing for the conflicting conditions. We compute the set $CC = conflicting\_conditions(m)$ which returns the set of conflicting conditions with that particular medicine. We then check for conflicts by taking $CC \cap Cp = \emptyset$. If condition is true, we update this information for the patient by computing $CCp = CCp \cup CC$. Similarly we compute the set $CF = conflicting\_food(m)$ which will return the set of conflicting food items for that particular medicine. We then check for conflicting food items by computing $CF \cap Fp = \emptyset$. If this set is empty then no conflict is found and an updated version of the conflicting food items is computed $CFp = CFp \cup CF$. The next section will show how to check for conflicts at each component of the system.

5 MISS Conflict Checking

MISS is composed of three subsystems: The Doctor Subsystem, the Pharmacy Subsystem and the Smart Home Subsystem. MISS will also access a global medicines database from a trusted third party. Each of these subsystems is responsible for checking for conflicts but it is expected that each one captures a more specific set of conflicts than the others. The algorithm for checking conflicts is very similar. Therefore we define the following two routines as follows:

Definition: Get Data (GD)

Input: Prescription *r* = *(p, m)*

//Get the data of *p* querying the local database

Query *Mp, Cp, Fp, CMp, CCp, CFp*

//Compute data of *m* from the global medicines database

Compute CM, CF, CC

Definition: Conflict Checking (CC)

1. Input: Prescription *r* = *(p, m)*
2. Call Get Data (r)
3. If ($CM \cap Mp = \emptyset$ and $m \cap CMp = \emptyset$)
4. If ($CC \cap Cp = \emptyset$)
5. If ($CF \cap Fp = \emptyset$)
6. //No Conflict found
7. $CMp = CMp \cup CM$
8. $CCp = CCp \cup CC$
9. $CFp = CFp \cup CF$
10. Else
11. Medicine *m* creates food conflict
12. Else
13. Medicine *m* creates a health condition conflict
14. Else
15. Medicine *m* creates a medicines conflict

5.1 Conflicts at the Doctor Subsystem

In this model it is assumed that the process starts when the patient visits the doctor and is prescribed with some medicine. Therefore at the Doctor Subsystem *(DS)* the model is fed with a new prescription *r* = *(p, m)*. This prescription *r* contains the id of the patient *p* and the id of the prescribed medicine *m*. The DS will have a local database with information stored about the patient *p* such as previously prescribed medicines, and health conditions. The DS is assumed not to store any information about food, so this set will be empty. This means that in the DS checking for conflicting health conditions and conflicting medicines will be enforced. This check will be performed by invoking the previously defined function *CC* with input *r*. If no conflict is found then the prescription *r* is sent to the Pharmacy Subsystem for further checking.

5.2 Conflicts at the Pharmacy Subsystem

At the Pharmacy Subsystem *(PS)* it is assumed that the prescription *r* has been checked at the *DS* and no conflict has been found. The *PS* therefore will receive the prescription *r* from the doctor. It will then use this information to further check for conflicts. It is expected that the *PS* will have a local database with the patient's record of previous prescriptions and over-the-counter medicines bought at that pharmacy. This data may be different to the one at the *DS*. It is possible that the patient is visiting different doctors and a different prescription from different doctors might be the source of conflict. The patient also might buy over-the-counter medicines which could be the ones that create the conflict, so all of them must be checked. It is assumed that the pharmacy does not store information about food, so this set would be empty. Therefore the *PS* must check again for conflicting conditions and conflicting medicines but using the pharmacy's local data. This is performed by calling the previously defined function *CC* with input *r* using the pharmacy's dataset. If no conflicts are detected the data is clear to be sent to the Smart Home Subsystem.

5.3 Conflicts at the Smart Home Subsystem

The Smart Home Subsystem (*SS*) will receive the data from the *PS* in the form of a prescription *r*. The *SS* is expected to check for any remaining possible conflict among medicines such as those picked up at different pharmacies. We are expecting the *SS* to have a local database with an inventory of the medicines and the food items available. This will allow checking if there is any remaining medicines conflict. The patient might be visiting different pharmacies and different doctors. The medicines prescribed from different doctors might create a conflict at this should be detected at the pharmacy. But if the patient is also visiting different pharmacies, these conflicts can go undetected. These multiple paths of conflicts are the ones that the *SS* will be responsible of detecting. Also again the patient might buy over-the-counter medicine at a gas station or grocery store for example. These can create undetected conflicts as these stores are not part of our system. But when the patient arrives at the Smart Home, the *SS* have the capability of detecting these conflicts with over-the-counter medicines as well. The *SS* also detects any food items in conflict with the new prescription or with over-the-counter medicines. All these checks are performed by calling the function *CC* with input *r* using the *SS* local dataset which includes the medicines inventory of previous prescriptions and over-the-counter medicines, and the set of food items available.

6 Instantiation of the Model and System Design

Now that we have defined the main subsystems we want to ensure that everything works correctly and the system actually detect conflicts. For this instance imagine the following scenario. A patient visits the doctor and the doctor prescribes three medicines. The doctor records indicate a previously prescribed medicine and diagnosed health condition. One of the three newly prescribed medications will create a conflict and this will be detected. Later the patient goes to the pharmacy, where he previously picked up a prescription from another doctor. The system should detect a conflict among the medicines prescribed by these two different doctors. Now the patient arrives home and only one of the three prescriptions so far has not find any conflict. But at the smart home the patient has some food item which should be avoided with that medicine and this is detected by the system. Based on the previous scenario we will present now an abstract instance of the model followed by an instance that uses real data of drug interactions pulled from the PDR Drug Interaction Tool [15].

Let's consider an abstract instance of the model. In *Table 1* we have the patient's data stored at the doctor's module in the row DS, the patient's data stored at the pharmacy's module in the row PS and the patient's data stored at the smart home module in the row SS. *Mp, CMp Cp, Fp* are as defined in Section 5. The medicines *MA, MB* and *MC* are prescribed by the doctor with the data as shown in *Table 2*. CM, CF, CC are as defined in section 5. In this instance we have that for prescribed medicine *MA*, the *DS* will detect a conflict with conditions *C1*, but medicines *MB* and *MC* will find no conflicts based on the doctor's data about the patient. When prescription data arrives to the pharmacy, the *PS* will detect a medicines conflict among prescribed medicine *MB* and previously prescribed medicine *M2*. No conflict is found with medicine *MC* up to this point. When the *SS* checks, it finds a food-drug conflict with prescribed medicine *MC* and food item *F3*. Therefore the patient or caregiver can be informed of this. This is an example on how the lack of information of the entire context can lead to a conflicting prescription, but with this system it will eventually be detected. The abstract model shows that this can be applied to any set of medicines. Also shows an example of the different paths that can create a conflict. In general a patient is seeing different doctors who prescribe different medicines. Prescriptions from one doctor might create a conflict with prescriptions from another doctor and the pharmacy module would detect that. But if the patient is also visiting different pharmacies these conflicts may go undetected. Also buying over-the-counter medicines at places different than the pharmacy can create the conflict. But at the Smart Home subsystem these conflicts would be detected as it works as a sink node. We want to ensure safety in this system which is one of the main motivations to perform these conflict checks repeatedly from the very beginning of the process.

Consider now the case with real drug interaction data set pulled from the PDR Online Drug Interaction Tool. In Table 3 we have the data stored at the doctor's module in the row DS, the data stored at the pharmacy's module in the row PS and the data stored at the smart home module in the row SS. Table 4 shows the medicines prescribed by the doctor in this case Zoloft, Percocet and Allegra. The data about the conflicts is also presented in each column. At the doctor's module the patient's data indicates he was previously prescribed Ambien and has a condition of hallucinations. Therefore a conflict is detected with the prescribed medicine Zoloft which is not recommended if a patient has hallucination and also conflicts with Ambien. The rest of the prescription Percocet and Allegra find no conflict with Ambien. At the pharmacy the patient previously had a prescription of Xanax from a different doctor. Therefore a conflict with Percocet is detected as these two medicines should not be taken together. The medicine Allegra has found no conflict yet. Now at the Smart Home module we have the medicines inventory and the food inventory. A conflict is found between Allegra and Orange Juice as they should not be taken together. In the next subsections the prototyped implementation of this model are described.

Table 1. Patient's data at each subsystem

	Mp	**CMp**	**Cp**	**Fp**
DS	M1	M11	***C1***	F1
PS	**M2**	***MB***	C2	F2
SS	M1, M2, M3	M11, MB, M33	C1, C2, C3	F1, F2, ***F3***

Table 2. Prescriptions at DS

M	**CM**	**CC**	**CF**
MA	M0	***C1***	F11
MB	***M2***	C22	F22
MC	M32	C33	***F3***

Table 3. Patient's medicines at each subsystem

	Mp	**CMp**	**CCp**	**Fp**
DS	***Ambien***	Zoloft	***Hallucinations***	*
PS	Xanax	***Percocet***	*	*
SS	Ambien, Xanax	Ambien, Percocet	*	***Orange Juice***

Table 4. Example of a prescription at DS

M	**CM**	**CC**	**CF**
Zoloft	***Ambien***	***Hallucinations***	*
Percocet	***Xanax***	*	*
Allegra	*	*	***Orange Juice***

7 Prototype Implementation

The prototype implementation of several use cases in our Smart Home Lab shows the feasibility of this system. The Doctor, Pharmacy and Smart Home Subsystems have been implemented as follows. One node has a customized application for inputting the prescription details and stores these in a database acting as the doctor's module. This node sends the data to another node which acts as a server with the patient dependent and patient independent information corresponding to the pharmacy. It assigns an RFID tag to the prescription received from the doctor's module. Then we have another node acting as the client, querying the pharmacy's computer using an RFID tag as the key. We used a Phidget RFID reader [16] and assigned different RFIDs to several containers and tested the reading of tags. After reading the tag the client computer queries the pharmacy database and downloads the specific information that matches the RFID-tag as the primary key. The information includes details about the prescription. This data obeys a format that the Smart Home can store it in its database and use it to update other subsystems such as the reminder, notification and medicine inventory. When a conflict is detected the corresponding message is displayed. The Smart Home subsystem was developed as a bundle that runs is OSGi, a framework particularly suitable for Smart Home applications [17]. The flow of data was correctly transferred from the Doctor's node all the way to the Smart Home and the experimental conflicts were correctly detected.

8 Conclusions and Future Work

The management of medicines and prescriptions by the elderly and people with special needs might be a challenging task for this population. A system that reduces manual intervention and a model for checking and detecting conflicts is presented. This system starts with a visit to a doctor which enters the prescription information into the system and check for conflicting medicines and health conditions. This data is made available to the pharmacy, through RFID-enabled prescriptions. The pharmacy performs a double check for conflicting medicines and health conditions and prescriptions are dispatched in special RFID-enabled containers. The patient scans the special containers in the Smart Home which updates the system with the data it downloads from the pharmacy. The Smart Home checks for conflicting medicines, health conditions and food items and if no conflict is found updates the calendar, reminders, inventory and other subsystems. MISS, the system described in our paper correctly detects conflicts using a formal model and will facilitate the task of giving reminder and increasing medicine in-take compliance. Integrating this system with the whole medicine dataset provided by the official governmental and medical entities such as the FDA and PDR is on-going.

References

1. Wan, D.: Magic medicine cabinet: A situated portal for consumer healthcare. In: Proceedings of the International Symposium on Handheld and Ubiquitous Computing, Karlsruhe, Germany (1999)
2. Siegemund, F., Flörkemeier, C.: The Smart Medicine Cabinet
3. Brusey, J., Harrison, M.: Reasoning about uncertainty in location identification with RFID. In: IJCAI 2003 (2003)
4. Floerkemeier, C., Lampe, M., Schoch, T.: The Smart Box Concept for Ubiquitous Computing Environments. In: Smart Objects Conference (sOc) 2003, Grenoble, France (May 2003)
5. Siegemund, F., Floerkemeier, C.: Interaction in Pervasive Computing Settings using Bluetooth enabled Active Tags and Passive RFID Technology together with Mobile Phones. In: PerCom, Fort Worth, USA (2003)

6. Lampe, M., Flörkemeier, C.: The Smart Box Application Model (2004)
7. Nugent, C., Finlay, D., Davies, R., Paggetti, C., Tamburini, E., Black, N.: Can technology improve compliance to medication? In: a.H.P., S.G. (ed.) From Smart Homes to Smart Care, pp. 65–72. IOS Press, Amsterdam (2005)
8. Szeto, A., Giles, J.A.: Improving oral medication compliance with an electronic aid. IEEE Engineering in Medicine and Biology Magazine 16, 48–54 (1997)
9. Vergnès, D., Giroux, S., Chamberland-Tremblay, D.: Interactive Assistant for Activities of Daily Living. In: 3rd International Conference on Smart Homes and Health Telematics, ICOST 2005, July 4-6, 2005, Sherbrooke, Canada (2005)
10. Paggetti, C., Tamburini, E.: Remote management of integrated home care services: the dghome platform. In: Press, I. (ed.) From Smart Homes to Smart Care: ICOST 2005 (2005)
11. Pill Dispenser CompuMed. (Accessed February 11, 2008), http://www.epill.com/
12. Med-ic. Home Page (Accessed February 11, 2008), http://www.med-ic.biz/
13. Foo, V., Fook, S., Haur Tee, J., Sang Yap, K., Aung Phyo Wai, A., Jayachandran, M., Biswas, J., Hin Lee, P.: Smart Mote-Based Medical System for Monitoring and Handling Medications. In: 5th International Conference On Smart Homes and Health Telematics, ICOST 2007(2007)
14. FDA. Home Page (Last accessed Feb 11, 2008), http://ww.fda.gov
15. Physician Desktop Reference. Home Page (Last accessed Febuary 11, 2008), http://www.pdrhealth.com/
16. Phidgets. Home Page (Last accessed February 11, 2008), http://www.phidgets.com
17. Gu, T., Pung, H.K., Zhang, D.Q.: Toward an OSGi-Based Infrastructure for Context-Aware Applications. In: IEEE Pervasive Computing, October 2004, vol. 3(4), pp. 66–74 (2004)