



中央研究院  
資訊科學研究所

Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-07-004

## Point-of-Care Support for Error-Free Medication Process

J. W. S. Liu, C. S. Shih, P. H. Tsai, H. C. Yeh, P. C. Hsiu, C. Y. Yu, and  
W. H. Chang



February 26, 2007 || Technical Report No. TR-IIS-07-004

<http://www.iis.sinica.edu.tw/LIB/TechReport/tr2007/tr07.html>

Institute of Information Science, Academia Sinica

Technical Report TR-IIS-07-004

# Point-of-Care Support for Error-Free Medication Process

J. W. S. Liu, C. S. Shih, P. H. Tsai, H. C. Yeh, P. C. Hsiu, C. Y. Yu, and W. H. Chang

## Abstract

Technological advances and critical needs have led to a variety of smart devices designed for prevention and reduction of medication errors. The focus of this paper is on devices and tools that support medication administration for this purpose. They include smart medication carts, robots and dispensers for professionals, as well as smart dispensers for naïve users. The paper describes device architectures, interfaces and support environment needed to increase the effectiveness of such devices in prevention of medication errors, as well as missing standards that will enable their integration in medication process tool chains and their wide spread use.

Copyright @ February 2007

---

J. W. S. Liu is affiliated with Institute of Information Science, Academia Sinica, Nankang, Taipei, Taiwan. Her email address is [janeliu@iis.sinica.edu.tw](mailto:janeliu@iis.sinica.edu.tw).

C. S. Shih, H. C. Yeh, P. C. Hsiu, and W. H. Chang are affiliated with Department of Computer Science and Information Engineering, National Taiwan University, Taiwan. Their email addresses are {cshih, r94922048, r91004, xx}@[csie.ntu.edu.tw](mailto:csie.ntu.edu.tw).

P. H. Tsai and C. Y. Yu are affiliated with Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. Their email addresses are [peipei@iis.sinica.edu.tw](mailto:peipei@iis.sinica.edu.tw) and [rayswin@gmail.com](mailto:rayswin@gmail.com).

## Table of Contents

Abstract.....	1
Table of Contents .....	2
1 Introduction.....	3
2 An Environment for Minimization of Medication Error.....	4
3 Medication Schedule Specification.....	8
4 Prescription Authoring Tool.....	12
5 Automated Medication Dispensers .....	17
6 Summary and Future Work .....	22
Acknowledgements.....	23
References.....	23

## 1 Introduction

Recent literature on health care quality (e.g., [1-6]) contains numerous alarming statistics on occurrences and consequences of preventable medication errors. As examples, estimated numbers of preventable adverse drug events each year in US hospitals, in long term care facilities and among Medicare patients are approximately 400,000, 800,000 and 500,000, respectively. Medication errors can occur at any stage of the use process, from ordering, transcription and dispensing to administration. Administration errors, leading to non-compliance to correct medication directions, contribute a significant percentage (25 – 40%) of all errors. Non-compliance in US ranges from 15 to 50 %, even for medications used to treat serious chronic conditions. It is the cause of 25 % of admissions to nursing homes and has led to billions of dollars in hospital cost and deaths in thousands each year [5].

In the hands of health-care professionals, medication administration has been labor intensive [6] and, hence, can be error-prone. A naive user may be on multiple interacting medications and have many prescriptions each year, in addition to over the counter (OTC) drugs and health supplements. Some medications have complex regimens and some for long durations. Staying compliant over years and decades is challenging, even for well educated and disciplined users. These facts have motivated many proposals and projects on use of device and information technologies for prevention of medication errors and improvement in compliance (e.g., [7-14]). Numerous administration and compliance aids have become available or are emerging. For health-care professionals, pharmaceutical distributors and hospital equipment industry offer all sorts of medication carts, cabinets and robots [15] for administering medications in a controlled manner and tracking the use and inventory of medications managed by them. Devices and tools for personal and small clinic use range from pill boxes, alarms, and programmable medicine dispensers to on-line scheduling and time table formatting tools and services (e.g., [16-20]).

Existing administration and compliance-aid devices and tools, including entries of a 2006 design competition [20], fall short in many aspects; automation and integration are the most important. Take existing dispensers for personal use as an example. Some of the most frequently cited reasons for non-compliance include inability to understand directions and inconvenience of rigid schedules. Existing dispensers are of little or no help there: One must manually load the medications into the dispenser, understand their directions and have the device programmed accordingly. Modern drug libraries (e.g., [21]) provide comprehensive information on firm and hard dose size and timing constraints and compliance criteria for common medications. Because such information is not captured in a form that can be used to guide their operations, existing

dispensers and scheduling tools cannot take advantage of the information to make medication schedules more flexible and user-friendly, while keeping compliance enforcement rigorous.

This paper focuses on the point-of-care and end-user level of an open environment of information systems, tools and devices for integration and automation of medication process. It is encouraging to see increasingly wide spread deployments of some parts of such environments, including electronic health record [22], electronic medication administration record (eMAR) and prescription order entry systems (POES) [23-26]. By and large, they are not fully integrated with administration tools, however. For example, smart medication carts, cabinets and robots are common today in hospitals, as well as home-care and assisted living facilities. Most of them are custom built and have proprietary interfaces. Like medication dispensers for personal use, the tool environment does not support their integration and automation to the degree required for error-free medication administration [27]. Even the integration of devices with interface capabilities conforming to HL7 standards requires extensive custom development [15].

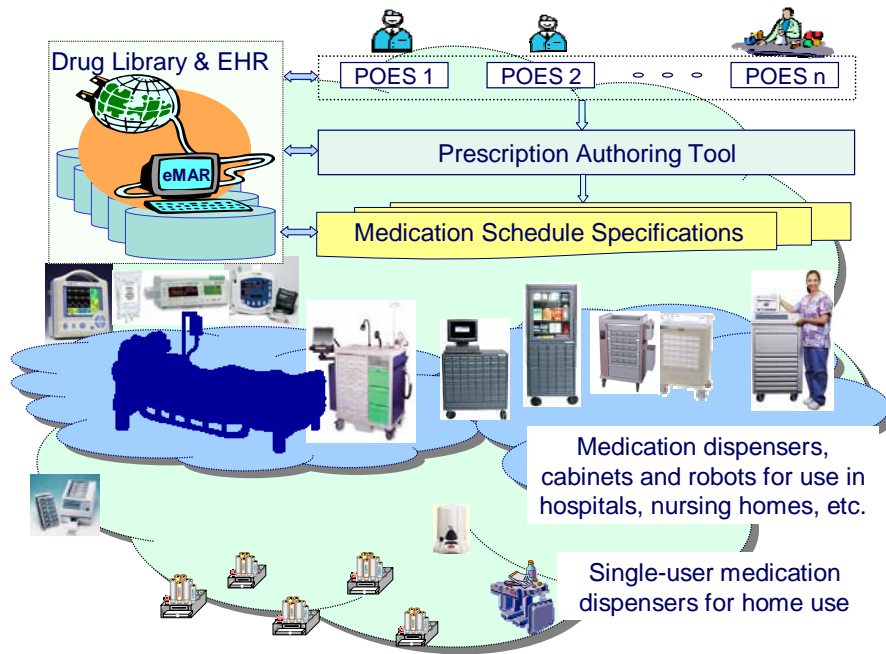
In the remainder of the paper, Section 2 provides further motivation for an integrated chain of medication process support tools that extend all the way to end-user administration devices and compliance aids. The underlying assumptions on the usage of these devices are summarized. Section 3 discusses the elements of machine-readable specifications required to define for automatic medication dispensers firm and hard constraints of dispensing schedules and compliance conditions. Section 4 describes a preliminary version of a transcription tool, called *prescription authoring tool* in [28], for transforming electronic prescriptions, on-line instructions and printed medication directions into such specifications. Section 5 describes a design of automatic dispensers for naïve users [29] and extensions and infrastructure support needed for the integration of smart medication dispensers for professional users. Section 6 summarizes the paper and discusses missing standards and future work.

## **2 An Environment for Minimization of Medication Error**

An open system of devices and tools designed to enhance automation and reduce errors of medication process is likely to resemble the one depicted in Fig. 1. All parts of the system are linked by one or more networks. Bold arrows between some parts indicate information flow paths; they are shown explicitly to emphasize close collaboration between the parts.

***Professional Administration Tools*** The tool chain in upper half of the diagram resembles the automated medication administration loop recommended in [27] for hospitals and health-care and assisted living institutions. Like that loop, the tool chain here starts from order entry system(s) via which medications for each patient in such an institution can be ordered.

Dispensing and administration tools down stream ensure on-time deliveries of the right numbers of unit doses of the right medications to the right locations. Once there, point-of-care tools can help care providers to administer the medications to the right patients in verifiably right manners.



**Fig. 1 Devices and tools for error-free medication process**

The loop shown here provides two forms of feedback: The data paths connecting electronic record systems with tools that generate medication directions and tools that administer medications enable timely updates of the electronic medication administration record of each patient and, thus, close the loop. Medical devices and monitors, exemplified by devices shown in the middle-left part of the diagram, provide another feedback path: Data on the condition of each patient captured by bedside monitors and devices are fused together not only for integrated displays, bedside and remote, but also for updates of the individual health and medication administration records. Because of the growing library of medical device interfaces that can transform data in proprietary formats into a standard format [30], fusion of data generated by devices from different vendors is now economically viable. This feedback provides the crucial information required to support notification mechanisms that can trigger timely revisions in medications of patients who fail to respond positively to current treatments.

**Tools for Non-Professionals** A system of tools for minimization of medication errors should also include personal medication dispensers and compliance aids for use by naïve users without close professional supervision. The devices at the bottom of the diagram in Fig. 1 intend to illustrate them. Smart medication dispensers for personal use are increasingly more essential to

ensure medication safety. Today, patients are typically discharged from hospitals while still on critical medications. Thanks to modern medical therapies and wonder drugs, many previously life-threatening diseases and chronic conditions can now be treated and controlled. A consequence is that increasingly more people are on medications on long term basis now than decades past. Other factors contributing to the increase in number and diversity of people who take medications at home and work include rapidly aging population and widely available OTC drugs and health supplements.

Today's personal smart dispensers are not as effective a tool for combating medication errors as they can be. As stated in previous section, poor usability is an issue. That aside, a major reason is that they are not sufficiently integrated with ordering and transcription tools upstream. Subsequent sections will discuss an approach to their integration and interfaces and standards required to make integration feasible and practical.

However, many of the challenges in incorporating personal medication dispensers into a medication process tool chain are non-technical. Some are concerned with usage restrictions, some with support infrastructure, some with financial incentives, and so on. An in-depth discussion on these aspects is out of scope of this paper. For the sake of discussion here, we make the following assumptions on how personal medication dispensers are used:

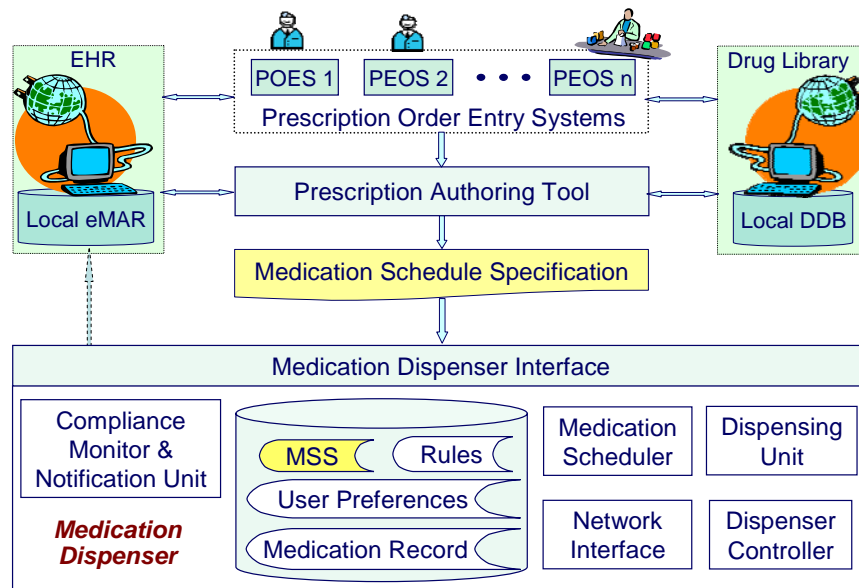
- (1) Each personal medication dispenser serves only one user, and the device manages all the medications taken by the user.
- (2) The user allows a complete and current electronic medication administration record (eMAR) according to some standard(s) to be maintained, and the record is accessible by the tool used to generate medication directions for the dispenser.

The restriction of single user per dispenser is imposed for sake of simplicity and can be easily removed. The restrictions that the device manages all medications of the user and the specification governing its operations is based on complete knowledge of all medications taken by the user are necessary for the device to be effective. Here, and in subsequent discussion, we use the term medication to mean a prescription drug, a OTC drug, some health supplement or food when there is no need to distinguish them. A dispenser does not handle food, clearly, but it must plan for meals and snacks along with medications when food interferes with some of the user's medications. A possible scenario is that the user is given the dispenser by a care provider (e.g., pharmacy, hospital or medical clinic). By allowing the device itself or the care provider institution maintain a complete medication history and thus enabling verifiable compliance, the user is rewarded with a reduction in medical insurance rate and the expense of the device covered in part or in full by insurance.

***Interfaces with Upstream Tools*** Again, our focus here is on medication administration. Thus

far, we have referred to tools designed to support this medication-process stage by a variety of common names, including smart medication carts, robots, cabinets, and dispensers. Hereafter, we call them all *medication dispenser*, or simply *dispenser*, when it is not necessary to be specific. We consider here only dispensers that support automation to a great degree. *Professional dispensers* are for health- and medical-care professionals and, typically, are used within care providing institutions. A *personal dispenser* is for use by an individual, at home or work and the person is not closely monitored by professionals in normal situations.

Fig.2 shows the connections from prescription order entry systems (POES) and electronic health record (EHR) systems upstream to a dispenser. The *prescription authoring tool* [28] in the diagram is an example of transcription tools that translate prescribed medication directions into administration instructions. We will discuss the role this and similar tools play in the next section. The block diagram at the bottom is that of a fully automatic personal dispenser. It is the subject of discussion of Section 5.



**Fig.2 Connection from upstream tools to dispenser**

Among the issues relating to interfaces between upstream tools and dispensers, the most important one is concerned with the information passed to each dispenser. Figs.1 and 2 call the machine-readable document containing this information a *medication schedule specification*, and *MSS* for short. The specification defines for the dispenser the rules that guide its operations. They include constraints on size and timing imposed on each dose to be administered, conditions to be met to stay compliant, and required actions to be taken when non-compliance occur.



### 3 Medication Schedule Specification

There have been very little open discussions to date on what needs to be defined by a medication schedule specification. This is less of an issue for professional dispensers. The contents of MSS for dispensers used by an institution are dictated not only by the directions of medications and the conditions of the patients involved, but also by policies, processes, practices, and operations of the institutions. One can readily justify the customization of such specifications to the institution, even to the individual departments within the institution. In contrast, an open standard for medication schedule specifications is needed to enable the wide spread, safe, and economical use of personal dispensers within a country and across countries worldwide.

To provide a rationale for our choice of the kinds of information a good MSS for a user of a personal dispenser or a patient cared by a professional dispenser should maintain, we note that the quality of a MMS can be measured by the quality of schedules produced by a dispenser following the specification. We measure the quality of a schedule along two dimensions: correctness and user-friendliness [29]. A schedule that governs time and quantity of each dose of every medication is correct when it rigorously compliant to all directions. It is user friendly if it fits the user's daily routine: A friendly schedule provides the user with sufficient flexibility by allowing deviations from the schedule without risking non-compliance. As examples, we sometimes see directions such as "take one tablet every 4 hours" and "take one dose with every meal and at bed time". A verifiably correct schedule can be easily generated according to the former, but the schedule is hard to follow, especially by an active and busy user on a long term basis. The latter may be user friendly, but it may lead to non-compliance if doses should not be taken too close together or too far apart. A challenge is to be sure that relaxation in compliance rigor for the sake of user-friendliness allowed by a MSS does not lead to errors.

***Firm, Hard, and Software Constraints*** Fig.3 shows an illustrative example. The MSS in the figure contains only the *medication directions (section)*. The section is *mandatory*, since the specification is useless without it. Some constraints defined by medication directions are firm and some are hard. *Firm constraints* guide the normal operations of the dispenser. *Hard constraints* define for the dispenser compliance conditions. Whenever possible, the normal schedules computed by the dispenser are such that all firm constraints are met if every dose is indeed administered as scheduled. Unfortunately, deviations from normal schedule may occur now and then, and some may lead to a violation of a hard constraint. The dispenser treats such a violation as a *non-compliance event* and is required to take some specified action(s) (e.g., contact a care taker). Clearly, specifications on the actions required to handle each type of

non-compliance events should be included in the MSS. We omit this aspect here. We also omit soft constraints, which the dispenser tries to meet on a best effort basis. A smart dispenser may accept user input on preferred times and frequencies for medications. *Soft constraints* are defined by user preference parameters. The report in [31] provides further discussion on these matters.

<pre> 1 &lt;!-- XML definition of Medication Schedule Specification--&gt; 2 &lt;?xml version="1.0" encoding="ISO-8859-1" ?&gt; 3 &lt;medicationScheduleSpecification user= "John W. Lee" &gt; 4   &lt;medicationDirections&gt; 5     &lt;medication&gt; 6       &lt;name&gt; 7         &lt;generic&gt;Alendronate sodium&lt;/generic&gt; 8         &lt;brand&gt;Fosamax&lt;/brand&gt; 9       &lt;/name&gt; 10      &lt;directionSources&gt; 11        &lt;prescriptionNo&gt;886919363344-72&lt;/prescriptionNo&gt; 12        &lt;drugLibrary&gt;PDRHealth&lt;/drugLibrary&gt; 13      &lt;/directionSources&gt; 14      &lt;dosageParameters&gt; 15        &lt;doseSize granularity="5 mg tablet"&gt; 16          &lt;nominalDoseSize&gt; 17            &lt;min&gt;2&lt;/min&gt; &lt;max&gt;2&lt;/max&gt; 18          &lt;/nominalDoseSize&gt; 19          &lt;absoluteDoseSize&gt; 20            &lt;min&gt;2&lt;/min&gt; &lt;max&gt;14&lt;/max&gt; 21          &lt;/absoluteDoseSize&gt; 22          &lt;budget&gt;14&lt;/budget&gt; 23          &lt;minTotalIntake&gt;7&lt;/minTotalIntake&gt; 24        &lt;/doseSize&gt; 25        &lt;timeParameters timeUnit="30 minutes"&gt; 26          &lt;duration&gt; 27            &lt;min&gt;8760&lt;/min&gt; &lt;max&gt;17520&lt;/max&gt; 28          &lt;/duration&gt; 29          &lt;nominalSeparation&gt; 30            &lt;min&gt;36&lt;/min&gt; &lt;max&gt;48&lt;/max&gt; 31          &lt;/nominalSeparation&gt; 32          &lt;absolutionSeparation&gt; 33            &lt;min&gt;36&lt;/min&gt; &lt;max&gt;336&lt;/max&gt; 34          &lt;/absolutionSeparation&gt; 35          &lt;replenishmentTime&gt;336&lt;/replenishmentTime&gt; 36          &lt;minIntakeInterval&gt;336&lt;/minIntakeInterval&gt; 37        &lt;/timeParameters&gt; 38      &lt;/dosageParameters&gt; 39      .... 40    &lt;/medication&gt; 41  &lt;/medicationDirections&gt; 42  &lt;!-- directions for other medications --&gt; 43 &lt;/medicationDirections&gt; 44 &lt;/medicationScheduleSpecification&gt; </pre>	<pre> 1 &lt;medicationScheduleSpecification user= "John W. Lee" &gt; 2   &lt;medicationDirections&gt; 3     &lt;medication&gt; 4       &lt;name&gt; 5         &lt;brand&gt;Fosamax&lt;/brand&gt; 6       &lt;/name&gt; 7       &lt;dosageParameters&gt; 8         &lt;!-- dosage parameters of Fosamax --&gt; 9       &lt;/dosageParameters&gt; 10      &lt;specialInstructions&gt; 11        &lt;sources&gt;PDRHealth&lt;/sources&gt; 12        &lt;drugAndFoodInteraction&gt; 13          &lt;separation time_unit="30 minutes"&gt; 14            &lt;interferer&gt; 15              &lt;name&gt;allDrugsAndFood&lt;/name&gt; 16              &lt;minToInterferer&gt;1&lt;/minToInterferer&gt; 17              &lt;minFrInterferer&gt;12&lt;/minFrinterferer&gt; 18            &lt;/interferer&gt; 19          &lt;/separation&gt; 20          &lt;changeLists&gt; 21            &lt;interferer&gt; 22              &lt;name&gt;Bayer Aspirin&lt;/name&gt; 23              &lt;absoluteDoseSize&gt; 24                &lt;min&gt;2&lt;/min&gt;&lt;max&gt;2&lt;/min&gt; 25              &lt;/absoluteDoseSize&gt; 26              &lt;absoluteSeparation&gt; 27                &lt;min&gt;36&lt;/min&gt;&lt;max&gt;48&lt;/min&gt; 28              &lt;/absoluteSeparation&gt; 29            &lt;/interferer&gt; 30          &lt;/changeLists&gt; 31          &lt;dietaryInstructions&gt; 32            &lt;takeWith&gt;water&lt;/takeWith&gt; 33          &lt;/dietaryInstructions&gt; 34        &lt;/drugAndFoodInteraction&gt; 35      &lt;/specialInstructions&gt; 36    &lt;/medication&gt; 37    &lt;medication&gt; 38      &lt;name&gt; 39        &lt;brand&gt;Bayer Aspirin&lt;/brand&gt; 40      &lt;/name&gt; 41      .... 42    &lt;/medication&gt; 43  &lt;/medicationDirections&gt; 44 &lt;/medicationScheduleSpecification&gt; </pre>
---	---

(a)

(b)

**Fig.3 An example of XML medication schedule specification**

As the example illustrates, the MSS for a user’s dispenser contains a section for each medication, illustrated by lines delimited by a pair of <medication> and </medication>. The space available in the figure allows us to show in the first segment (lines 6 – 13 in part (a)) of the section only the name of the medication and sources of the direction information. – Here, they tell us that the medication is Fosamax, which is a brand of a drug prescribed for prevention and treatment of brittle bone disease and is usually taken on a long term basis. The directions and instructions are based on the user’s prescription, as well as information extracted from the

on-line drug library PDRHealth [21]. – In general, this segment should provide information on all attributes (e.g., bar code or RFID, physical characteristics, and supplier id) that the dispenser needs to administer the medication.

The section for each medication has two parts: dosage parameters and special instructions. The *dosage parameters (DP)* part includes the direction for the named medication when the medication is taken alone. If the medication interacts with any other medication taken by the user or with food, then the section for the medication also has a *special instructions (SL)* part; it specifies how the administration of the medication is to be modified because of the interaction.

**Dosage Parameters (DP)** Lines 14 - 38 in the XML specification in Fig. 3(a) illustrate the dosage parameters part. The part defines firm and hard constraints that any schedule of the medication must satisfy. Firm constraints on dose size and separation between consecutive doses are given by *nominal dose size range* (lines 16 – 18) and *nominal separation range* (lines 29 – 31), respectively. The specification here indicates that the dispenser normally should schedule a dose of Fosamax consisting of two tablets, for a total of 10 mg, every 18 to 24 hours.

The constraints defined by other parameters in the DP part are hard. They include *absolute dose size range* and *absolute separation range*. By making these ranges larger than the respective nominal ranges when it is safe to do so, the MSS enables the dispenser to provide scheduling flexibility safely. This is true in the case of our example. (They are given by lines 19 – 21 and 32 – 34, respectively, in part (a).) We note that as lines 14 – 18 in part (b) indicate, the medication should be taken at least half an hour before food and any other drug. A busy, active user may not be able to wait that long for breakfast every day. The fact the absolute dose and separation ranges are [2, 14] and [36, 336], respectively, allows the user to skip the normal 2-tablet daily dose now and then and take larger doses on some other days.

Allowing large ranges of dose size and separation, even occasionally, may lead to over dose or under dose. This is the reason for the inclusion of supply rate and demand rate constraints in the MSS. *Supply rate (B, R)* is defined by the parameters *budget B* and *replenishment time R*. It constrains the total size of all doses over any time interval of length *R* to be no more than *B*. For example, the direction of a pain killer may read “take 1 or 2 tablets every 4 to 6 hours, but never more than 6 tablets in 24 hours”. The supply rate of such a medication is  $(B, R) = (6, 24)$ . The supply rate of Fosamax in our example is given by lines 22 and 35, respectively. They say that the total size of doses within any 7 days can be at most 70 mg, i.e., 14 tablets.

Similarly, a *demand rate* constraint intends to ensure that a certain amount of the medication is at work at all times. This constraint is defined by two parameters: *minimum total intake L* and

*minimum intake interval P*: It requires that the total size of all doses in every interval of length  $P$  to be at least equal to  $L$ . These parameters are given by lines 23 and 36, respectively, in part (a): They say that the user must have at least 35 mg in any interval of 7 days.

Finally, the duration parameter specifies the length of time the medication is to be administered. In the case of Fosamax, the duration may be arbitrary. (Lines 26 – 28 specify that the duration is between six months and a year.) The constraint is important for some other medications, however, for many reasons. As examples, an upper bound to duration for a pain killer is imposed to prevent habit forming, and a lower bound for an antibiotic to ensure completeness of the treatment.

***Special Instructions (SL)*** Because drugs more or less interact, some modification in the user's medication regimen is often necessary when the user is on multiple medications, in order to keep the effects of interaction under control. There are two common types of direction modifications due to drug interaction: additional constraints on separations between doses and changes in dosage parameters of interacting medications. A MSS captures the information on these types of modification in the SL section of each interacting medication.

Lines 10 – 35 in part (b) of Fig.3 illustrate the SL part in a MSS. We call interacting medications here *interferers*. The SL section for each medication  $M$  that has one or more interferers has two parts: One part defines additional separation constraints, if any, and the other lists changes in its dosage parameters, if any. For each interferer for  $M$ , the *separation part* for  $M$  gives the required temporal separations between each dose of  $M$  and any dose of the interferer. The SL section of Fosamax in our example starts from line 10 and ends at line 35 in part (b) of Fig. 3. The separation part is listed starting from line 13. Since this medication should not be taken together with anything else, there is only one entry, allDrugsAndFood, in this part. Line 16 says the time from a dose of Fosamax to the time for anything else is at least half an hour, while line 17 says that the waiting time for the next dose of Fosamax is 6 hours if anything else is taken before the medication. That the required separation from the medication to an interferer differs from the required separation from the interferer to the medication is typical.

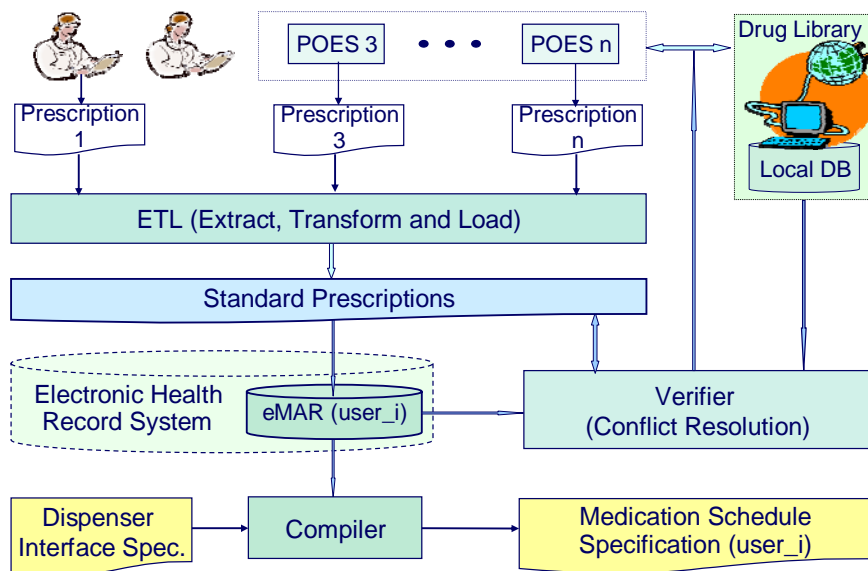
The *change lists part* has an entry for each interferer that affects  $M$  to the extent as to require changes in one or more dosage parameters of  $M$ . The parameters given by change lists take precedence over dosage parameters of  $M$  specified in its DP section as long as the user is taking both  $M$  and the interferer. In the case of Fosamax, its absolute dose size and separation ranges given in the DP section are [2, 14] and [36, 336], respectively. Hence a user taking the medication alone has the flexibility of skipping some doses and making up the missed doses later by taking a one with a larger size. The changes listed in lines 21 – 29 say that the dispenser

should no longer allow this flexibility as long as the user is taking aspirin as well.

We leave off many additional constraints due to medication interaction. Precedence constraint is one. A precedence constraint restricts the order in which doses of some interacting medications are taken. In addition to minimum separations imposed to minimize the effect of interaction, there may also be maximum separation constraints to ensure that some medications are taken sufficiently close together. Our report on a general model of medication scheduling [31] discusses these and other constraints, as well as a more abstract representation of medication interaction information than the way exemplified by Fig. 3.

#### 4 Prescription Authoring Tool

In Figs. 1 and 2, we call the tool that plays the critical role in the integration of medication dispensers with POES a *prescription authoring tool* [28]. It is the last line of defense against the propagation of prescription errors to administration tools. Fig.4 shows a general configuration of a prescription authoring tool [28].



**Fig.4 Structure of prescription authoring tool**

As we can see from the figure, the tool consists of four major components: ETL (Extract, Transform and Load) processor, compiler, verifier and, sometimes, a local database of medication records of users served by the tool. The input to the tool includes the user's (or patient's) prescriptions from order entry systems and directions for OTC drugs. They may be electronic, printed, or handwritten; in natural languages, specialized languages or tablet forms; and so on. Hereafter, we call them all prescriptions. The output is a MSS for the user's dispenser; the input device interface specification details the capabilities of the device.

***Use Scenarios and Operations*** An authoring tool that serves solely professional dispensers within a single institution may contain only the compiler. It makes sense for an institution to deploy standardized (at least, compatible) order entry systems and prescription schemas and vocabularies. Another reasonable requirement is that each prescription is made with near complete knowledge of the patient's medication record and all adjustments in directions to account for the effects of drug interactions were made at the time when each prescription is issued. When these assumptions are valid, the only required function of the authoring tool is to compile the user's prescription(s) into MSS for the dispenser(s) used there.

Authoring tools used by pharmacists to process medication directions for personal dispensers need all components in Fig.4. The assumptions stated above are typically invalid. The user of a personal dispenser may be under the care of independent care providers. They may have little or no knowledge of medications and health supplements ordered by others<sup>1</sup>, at least not until a universal health record system is established nation (or world) wide. Restriction (2) in Section 2 is imposed for this reason. Stated in another way, a requirement for correct use of automated dispensers is that the user's pharmacist has information on all prescriptions of the user, as well as information on user's OTC medications. This would be the case if the user gets all medication supplies from the same pharmacy. Whenever the user comes to fill a new prescription or to purchase a new OTC drug, the pharmacist uses the authoring tool to process the directions of the user's existing and new medications and generates a new MSS for the user's dispenser.

Errors introduced in the ordering stage (e.g., the ones described in [26]) are out of scope here. We start with the assumption is that each prescription is by itself correct: Someone or some tool has verified that the user can safely and effectively take the medications ordered by the prescription. However, some interactions among medications ordered by different prescriptions may not have been accounted for. It is the primary function of the authoring tool to make sure that the directions given by all prescriptions of the user are consistent and they are consistent with general directions and special instructions on drug interactions provided by drug libraries and medication advisory systems. The tool can resolve some inconsistencies based on common-sense rules provided to it for this purpose. A *conflict* is an inconsistency that tool cannot be resolved automatically based on the rules. A job of the tool is to bring all conflicts to the attention of responsible order entry systems and care provider(s) served by the systems.

The list below summarizes the work by the tool to process each prescription:

---

<sup>1</sup> This is especially true in countries where citizens have generous access to medical clinics, big and small, and physicians are willing to offer medications for minor ailments. For instance, it is no uncommon in Taiwan for an individual in reasonable health to visits doctors over 10 times a year. The user's health insurance IC card provides room for records of only six visits.

- (1) The ETL (Extract, Transform and Load) processor extracts directions from the new and existing prescriptions and puts the extracted directions in a unified representation.
- (2) The verifier checks for conflicts while merging user specific directions provided by each prescription together and with general directions and special instructions the tool extracts from drug libraries. If and when it detects some conflict, the tool sends a notification, containing information on the conflict, to the order entry systems and the physicians responsible for the conflicting prescriptions and requests conflict resolution. The verifier repeats (1) and (2) until all conflicts are resolved. At the end of the merging process, it stores the merged directions in the user's medication record and invokes the compiler.
- (3) When invoked, the compiler translates the merged directions into a machine-readable MSS for the user's dispenser. The compiler also carries out test to ensure feasibility of the MSS and for dispensers that are incapable of doing the work, compute one or more medication schedules for the dispenser.

We use the term ETL loosely here to mean the process of extracting directions from multiple prescriptions and drug libraries, and then transforming and merging them into one unified prescription. Information extraction should be straightforward to implement if standards for documents and forms, similar to business schemas standards for electronic commerce and supply-chain management and supported by many document processing tools, are commonly adopted for prescriptions. Without such standards, this step is ad hoc and error prone. The ETL processor in our authoring tool prototype can handle only prescriptions written in common and simple forms. We will add the capability of extracting directions from plain, easy-to-understand English descriptions, as exemplified by special instructions in [21].

**Rules and Algorithms** The authoring tool examines one prescription at a time, resolve any inconsistency if it can, and merges the prescription with already merged ones when the new prescription is deemed consistent. The rules for this purpose include the APAMAT prescription syntactic rules [28]. (APAMAT stands for prescription algebra for medication authoring tool.) Others are based on common-sense reasoning. As an example, Fig. 3(b) leaves out the change in supply rate constraint from (14, 336) to (2, 48). The tool can add the change since it can determine precisely the new parameter values from the changes in absolute dose size and separation ranges. Similarly, the tool can resolve differences in parameter ranges given by different prescriptions by using the most stringent one when the ranges intersect, unless it is instructed to do otherwise. In contrast, it must declare a conflict when the ranges are disjoint.

The demand-versus-supply test (DST) described by Fig.5 allows the tool to check nominal dosage parameters and rate constraints of each medication for consistency and feasibility [32].

The *consistency conditions* are listed in Step 1 in Fig. 5. The set of nominal dose-size range [ $d_{min}$ ,  $d_{max}$ ], separation range [ $s_{min}$ ,  $s_{max}$ ] and supply rate ( $B$ ,  $R$ ) and demand rate ( $L$ ,  $P$ ) parameters given by a MSS is *consistent* if the parameters satisfy the conditions listed there. The set, and the MSS, is *feasible* if there is a *feasible schedule* that meets all the constraints defined by the parameters.

**Demand-Versus-Supply Test (MD parameters of  $M$ )**

Input: dose-size range [ $d_{min}$ ,  $d_{max}$ ], separation range [ $s_{min}$ ,  $s_{max}$ ]  
supply rate ( $B$ ,  $R$ ) and demand rate ( $L$ ,  $P$ ) constraints,  
duration range [ $T_{min}$ ,  $T_{max}$ ]

Output: *feasible* = TRUE;  
*feasible\_schedule* = NULL;

1. Check input parameters for validity conditions listed below:
  - Valid ranges:  $0 \leq d_{min} \leq d_{max}$  and  $0 < s_{min} \leq s_{max}$
  - Valid limits:  $d_{max} \leq B$  and  $s_{max} \leq P \leq T_{min}$
  - Feasible supply rate:  $d_{min} \times \text{Floor}[R/s_{max}] \leq B$
  - Feasible demand rate:  $\text{Ceil}[L/d_{max}] \leq \text{Floor}[P/s_{min}]$
  - Consistent rate limits:  $B/R \geq L/P$
if any condition above is not met, *feasible* = FALSE; return;
2.  $L/P = 0$ , or  $B/R = \infty$ , return;
3. Carry out steps of one of the following cases:
  - if  $R \geq P$ : Find the *minimum\_demand\_schedule* and the *required\_supply* of the schedule;  
if *required\_supply* >  $B$ , *feasible* = FALSE;  
else *feasible\_schedule* = *minimum\_demand\_schedule*;
  - if  $R < P$ : Find the *maximum\_supply\_schedule* and the *available\_supply* of the schedule;  
if *available\_supply* <  $L$ , *feasible* = FALSE;  
else *feasible\_schedule* = *maximum\_supply\_schedule*;
return;

**Fig.5 Demand-Versus-Supply Test for Feasibility**

The conditions checked in Step 1 are necessary for the existence of a feasible schedule [29]. This is why Step 1 returns and reports that the MSS is not feasible if the parameters fail to meet any of the conditions. The consistency conditions are both necessary and sufficient for several special cases, one of which is that the medication does not have both supply rate constraint and demand rate constraint. This is the reason for Step 2 in Fig. 5

To define the terms used in Step 3 in Fig. 5, we note that a periodic schedule can be defined by the sequence  $\{(t_0, d_0), (t_1, d_1) \dots (t_i, d_i) \dots (t_{k-1}, d_{k-1})\}$  of 2-tuples of dose time  $t_i$  and size  $d_i$ .  $k > 0$  is the number of doses per period.  $t_0$  is the time of the first dose in a period. Subsequent doses are scheduled at times  $t_1, t_2 \dots t_{k-1}$  for  $t_0 < t_1 < \dots < t_{k-1}$ , relative to  $t_0$ . We call such a schedule a *k-dose schedule*. Let the total size of doses in each period be denoted by  $\delta(k)$ , and the length of the period by  $\pi(k)$ . Let  $t_k$  denote the end of the period.  $t_k = t_0 + \pi(k)$  is time of the first



dose of the next period; the size  $d_k$  of the dose at time  $t_k$  is  $d_0$ . Finally, let  $x_i = t_i - t_{i-1}$ , for  $i = 1, 2 \dots k$ , denote the separations between consecutive doses in each period.

For a given consistent set  $\{[d_{min}, d_{max}], [s_{min}, s_{max}], (B, R), (L, P)\}$  of dosage parameters, where  $R \geq P$ , a  $k$ -dose schedule is a  $k$ -dose *minimum demand schedule* (or *MinDS(k)* for short) when the dose sizes and separations in each period are solutions of the following problem:

***k*-dose Minimum Demand Problem:**

- (1) Find integers  $x_1, x_2 \dots x_k$  that maximizes  $\pi(k) = \sum_{1 \leq j \leq k} x_j$  subject to the constraints  $s_{min} \leq x_i \leq s_{max}$  for  $i = 1, 2, \dots k$  and  $\pi(k) \leq P$ .
- (2) Find integers  $d_0, d_1 \dots d_{k-1}$  that minimizes  $\delta(k) = \sum_{0 \leq j \leq k-1} d_j$  subject to the constraints  $d_{min} \leq d_i \leq d_{max}$  for  $i = 1, 2, \dots k$  and  $\delta(k) \geq L$ .

Similarly, for a consistent set of dosage parameters where  $R \leq P$ , a  $k$ -dose schedule is a  $k$ -dose *maximum supply schedule* (or *MaxSS(k)* for short) when the dose sizes and separations in each period are solutions of the following problem:

***k*-dose Maximum Supply Problem:**

- (1) Find integers  $x_1, x_2 \dots x_k$  that minimize  $\pi(k) = \sum_{1 \leq j \leq k} x_j$  subject to the constraints  $s_{min} \leq x_i \leq s_{max}$  for  $i = 1, 2, \dots k$  and  $\pi(k) \geq R$ .
- (2) Find integers  $d_0, d_1 \dots d_{k-1}$  that maximizes  $\delta(k) = \sum_{0 \leq j \leq k-1} d_j$  subject to the constraints  $d_{min} \leq d_i \leq d_{max}$  for  $i = 1, 2, \dots k$  and  $\delta(k) \leq B$ .

A *MinDS(k)* or *MaxSS(k)* schedule is *well-formed* if according to the schedule, the sizes of the doses differ by at most one, separations between consecutive doses differ by at most one, and if dose sizes are not equal or separations are not equal, larger doses are scheduled closer together earlier in the period. Because the coefficients in the objective functions in the problems above are equal, there is always a well-formed *MinDS(k)* (or *MaxSS(k)*) when the  $k$ -dose minimum demand problem (or the  $k$ -dose maximum supply problem) admits a solution.

The *required supply* of *MinDS(k)* is the total size of all the doses scheduled within the replenishment interval  $[0, R)$  when  $P \leq R < \infty$ . A *minimum demand schedule (MinDS)* of a MSS is a well-formed schedule that has the minimum required supply among all well-formed *MinDS(k)* for all feasible values of  $k$ . Similarly, the *available supply* of *MinDS(k)* is the total size of all the doses scheduled within the minimum intake interval  $[0, P)$  when  $R \leq P < \infty$ . A *maximum supply schedule (MaxSS)* of a MSS is a well-formed schedule that has the maximum available supply among all well-formed *MaxSS(k)* for all feasible values of  $k$ . A *MinDS(k)* (or *MaxSS(k)*) for each feasible value of  $k$  can be found in constant time. It follows that the minimum demand (or maximum supply) schedule of a specification can be found in time proportional to the maximum of dose-size and separation range widths.

As Fig. 5 indicates, if Step 3 succeeds, the test returns a minimum demand schedule (*MinDS*)

of the MSS when  $R \geq P$  or a maximum supply schedule (MaxSS) when  $R < P$ . Both schedules are periodic. We have shown that the specification has no feasible periodic schedule [32] if the step fails. (In other words, the test is optimal among all algorithms that produce periodic schedules.) However, the necessity of the test remains to be determined: We do not know whether the MSS has a non-periodic feasible schedule when Step 3 fails.

In principle, the schedule found by DST when the test succeeds can be used for selecting dosage (i.e., sizes and times of doses) of a medication. While the schedule is good for proving feasibility, it is deficient as a schedule, especially for a user who takes interacting medications. We measure the quality of a feasible dosage selection by *usable separation range (USR)*; it is the range of time allowed by the schedule without violating the given nominal range and rate constraints. The bigger the USR, the more flexibility the dosage selection gives to scheduling the medication with interacting medications and food. Because the DST test selects boundary values of constraint parameters, the USR of the schedule produced by it is typically zero. Several heuristic dosage selection algorithms presented in [29, 32] have lower success rates (e.g., between 40 to 55 % the success rate of DST) but can offer good USR when they succeed.

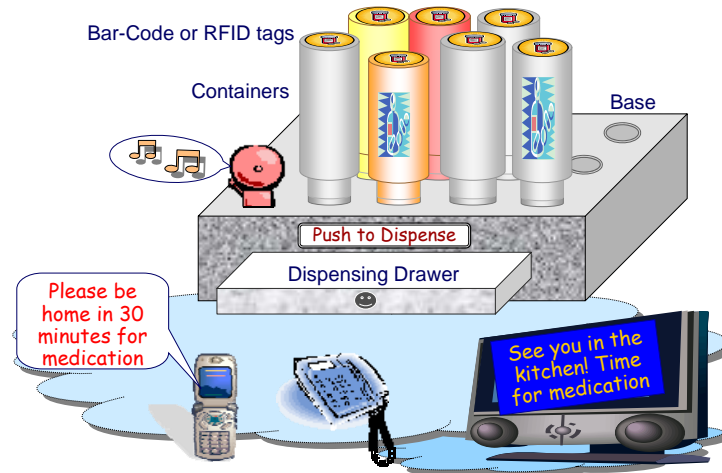
Thus far, we talked about tests for consistency and feasibility of dosage parameters of individual medications. APAMAT offers several simple rules for checking consistency of SL parameters of interacting medications. The set of consistency rules still needs to be expanded. The general problem of deciding whether a MSS is feasible is a hard problem, except for a few degenerate cases. We are developing simple polynomial-time algorithms and heuristic search schemes. Results generated by them are sufficient, but not necessary, for feasibility.

## 5 Automated Medication Dispensers

We are currently building a fully automatic personal medication dispenser [29]. Its block diagram is shown at the bottom of Fig. 2. In this section, we first discuss tradeoff issues that led to its design and briefly summarize its usage and operations. We then describe the usage assumption and design of a modular automatic professional dispenser.

*Some Tradeoff Issues* Fig. 2 shows that an automated personal dispenser has several key components: a medication scheduler, dispensing unit, dispenser controller, compliance monitor, and network interface. The dispenser also has a local database, allowing it to maintain, with user's permission, medication prescription and administration records, as well as data on user habits and behavior. Our dispenser has only a local alarm: The user is reminded that it is time to take medication(s) when the alarm sounds. The version does not require Internet access; a dial up

connection is adequate for notification purpose. In general, a dispenser in a modern home may support a variety of I/O devices as illustrated by Fig. 6: Then, reminders can be delivered to the user indoors and out, at home or away. A fully configured dispenser should have this option.



**Fig. 6 A fully automated personal dispenser**

One may want to ask why give the personal dispenser a scheduler? The authoring tool needs to compute some schedule(s) in the process of checking the MSS for consistency and feasibility. Then why not download those schedules to the dispenser along with the MSS. A minimal dispenser may work this way. However, a friendly dispenser must be able to dynamically adjust the medication schedule in response to changes in user preferences, to compensate for user tardiness, and so on. This is the reason for giving our personal dispenser a local scheduler. Even a powerful scheduler will not over stress the dispenser processor, and cost of the processor power and scheduler software is but a small part of dispenser cost.

An alternative is to let the dispenser use a scheduling service from some tool (e.g., the authoring tool, some other administration tool, or a more power dispenser.) This is a good set up for professional dispensers for many reasons. The scheduler for a professional dispenser may need to compute medication schedules for multiple patients and the schedules must fit the care provider's work routines. The large size of the scheduling and work planning problem may over stress the local processor of a low-end dispenser. A dispenser may be used to take care of different patients on different days. A better design is to have not only the MSS for the patients, but also their schedules, loaded at reassignment and initialization time. We chose this option for the automated professional dispenser design described shortly.

Fig. 6 shows that the container of each medication is labeled either by a bar code or a RFID tag. Our personal dispenser uses the latter. The use of bar-code labels is wide spread and their cost is significantly lower. However, the user must have the bar-code label of each container

scanned when the container is initialized. This manual step, and the associated potential for identification error, is eliminated by using RFID tags.

Finally, a dispenser cannot be sure that the user actually takes all the medication(s) in a sufficiently short time after the right doses is retrieved from the dispenser. This is a basic limitation of all personal dispensers without intrusive surveillance capability.

***A Personal Dispenser Design*** Fig.6 illustrates our dispenser. Every dispenser has a *dispensing unit*, which holds and dispenses medications. It is the component that interacts physically with the user. Its hollow base holds *dispensing mechanisms* (DM) and protects them from tampering. Medication containers are plugged in sockets on top of the base. The unit, as well as other components of the dispenser, is under the control of the *dispenser controller*.

To put new supplies of medications under the care of the dispenser, the user plugs the containers into the sockets, one at a time and one per socket. As stated earlier, each container comes from the pharmacy with an attached RFID tag. Each plug-in action causes the dispenser controller to read RFID tags and socket statuses and thus discover the location of the new container and the name of the medication in it. After making sure that the direction of the new medication is included in the MSS, the dispenser controller creates an association between the medication and its socket and maintains the association as long as the container remains plugged in. The controller also picks up from the MSS information on physical attributes of the medication. The DM responsible for the socket will need this information to operate the release mechanism. If the MSS does not contain the direction of the medication in a new container, the controller prompts the user for update of the MSS from a UBS flash memory provided by the pharmacy. The initialization process continues after the MSS is updated.

Shortly before it is time for the user to take a dose of any medication, the dispenser sends a *reminder* to the user. The user responds to the reminder by pressing the *Push-To-Dispense (PTD)* button on the dispenser. The dispenser controller waits until the PTD button is pressed. It then puts the size of the current dose of each medication that is due in the dose-size-register of the DM responsible for the release of the medication and commands the DM to place a dose of the specified size in the *dispensing drawer* below the base. The drawer is normally shut and stays shut as long as no dose is due. After the DM successfully places the dose in the drawer, the controller opens the drawer and presents to the user the doses due at the time.

The length of time the user takes to respond to a reminder is called *promptness*. The dispenser controller uses an estimate of promptness to determine when to send reminders. Promptness varies. The user may take longer than expected to respond from time to time. The directions of some medications call for skipping the current dose, or changing the current dose size, and so on, when user tardiness exceeds some threshold. The dispenser controller makes this

kind of adjustment when needed. This is why the controller waits until the PTD button is pressed before loading the dose-size-register of the responsible DM. The change in current dose size and time may require that subsequent doses of the involved and interacting medications be rescheduled. Since the dispenser has a local medication scheduler, it can easily re-adjust the schedule on the fly whenever the need arises.

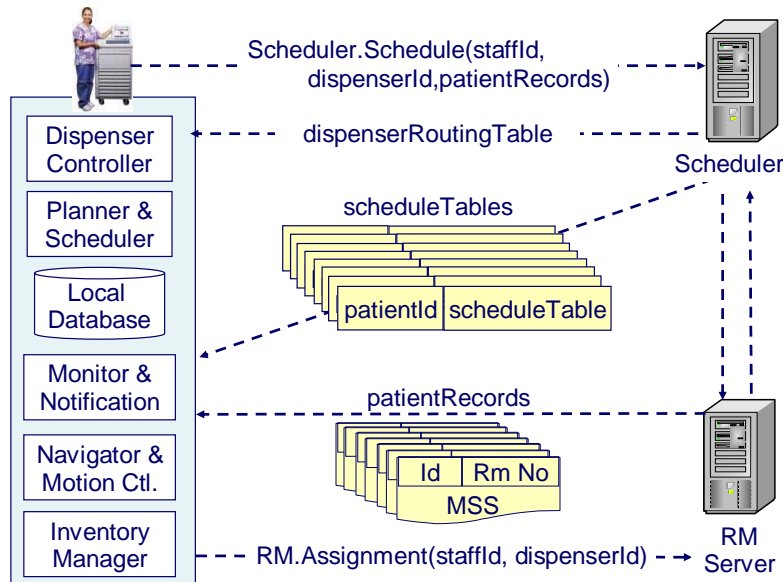
The medication scheduler is responsible for determining from the user's MSS and user preference parameters the time and dose size of each dose of every medication managed by the dispenser. It provides the dispenser controller with `Schedule( )` and `GetNextDose( )` functions. When called, the former produces a new `scheduleTable`, which is a list of `{time, doseList}` structures, sorted by time in increasing order. `time` in each entry gives the absolute time when a dose of some medication is due and `doseList` specifies the medications and their dose sizes to be dispensed at the time. The latter returns `{time, doseList}` at the head of `scheduleTable`.

The performance of a scheduling algorithm used by the dispenser is measured in terms of *success rate* and schedule quality. The former is the percentage time the algorithm succeeds in finding a feasible schedule for multiple interacting medications. Schedule quality has two dimensions: *percentage deviation* (from nominal constraints) and maximum allowed tardiness. Section 3 states that the dispenser prefers to use schedules that stay within the constraints of nominal dose size and separation ranges. Unfortunately, this is often not possible when there are several interacting medications to schedule. A schedule remains compliant as long as it never violates hard constraints defined by absolute parameter ranges and rate parameters. Nevertheless, a compliant schedule which deviates from nominal ranges a smaller percentage of the time is better. *Maximum allowed tardiness* is the maximum amount of time the user can be late in response to a reminder and still stay compliant. It quantifies the user friendliness of a schedule.

The scheduling algorithms we have studied so far are all priority basis [31]. There are two variants: non-greedy and greedy (priority-driven). A non-greedy algorithm computes the entire future schedule, for one medication at a time in a priority order. These algorithms generally have higher rates of success in finding feasible schedules. They also tend to produce better schedules. For a user who is often tardy to the extent to trigger rescheduling, it makes sense to compute the schedule on just-in-time basis using a priority-driven algorithm. Priority-driven algorithms tend to perform poorer, both in success rate and schedule quality.

***Professional Dispensers Architecture*** The goals of professional dispensers are to reduce not only medication errors but also the cost of medication administration. The block diagram in Fig. 7 shows the major components of an automatic professional dispenser for these purposes. A

scenario is that such a dispenser serves several to more than 10 patients in a nursing home or assisted living institution, depending on the amounts of attention they require. A single care provider (e.g., a nurse) oversees the administration of their medications. This person is the user of the dispenser. The dispenser automates everything, except the last step in administering each dose for each patient: By this statement, we mean that the dispenser determines time and size of every dose for every patient, navigates and moves to the right location, and places the right size dose at the scheduled dose time in a dispensing drawer at the location, and so on. In the last step, the care provider retrieves and verifies the dose, oversees that dose is taken by the patient and commands the dispenser to log the actual time. From the standpoint of dispenser design and implementation, this scenario is among the most complex ones. A dispenser serving one or two patients in intensive care does not need navigation and motion control, for example. Some automatic features (e.g., automatic dosage selection and adjustment) may also not be required.



**Fig. 7 Structure and support environment of professional dispensers**

The specifics about the infrastructure support system for dispensers are unimportant for our discussion here. It suffices to say that some components of the system provide resource management and scheduling services. Fig.7 calls the components RM Server and Scheduler. The former maintains information on patients (including records on their admissions, medications, and billing) and manages resources (including staff work assignment, dispenser usage, and medication inventory) of the institution. It may be a part of an enterprise resource planning system. The Scheduler provides scheduling and planning services for dispensers and is specifically for dispensers. It can be a part of a prescription authoring tool.

In the fully automatic mode, the dispenser sends a request for updates in work assignment information to the RM server whenever the user logs on the dispenser. The request contains the staff id of the user and the id of the dispenser. As a part of the response, changes in patients to be cared by the user and their medication records and physical locations are downloaded to the dispenser local database. Any subsequent change in this information triggers an update, either by the server or by the dispenser, depending on which one initiates the change. Because automatic update is not always desired, the dispenser also has a web based interface with the RM server, enabling such updates to be initiated and carried out manually by the user.

An automatic professional dispenser typically interacts closely with Scheduler. Fig.7 assumes that the dispenser sends the remote procedure call `Schedule( )` to invoke the computation of a new medication schedule. In this computation, the Scheduler first computes the medication schedules of individual patients based on their own MSS. It then coordinates the individual schedules whenever possible so as to minimize the overall time the care provider must spend to deliver and administer all of their medications.

As the figure shows, the result provided by the scheduler server consists of a `scheduleTable` for each patient. We measure the quality of these schedules also by percentage deviation and maximum allowed tardiness. The result also contains a `dispenserRoutingTable`. In essence, the dispenser routing table defines an overall medication schedule from the view point of the care provider. It is a list of structures `{time, patientId, location, doseList}`, sorted in increasing order by time. `time` gives the absolute time when the next dose of some medication is due, and `doseList` gives the names and dose sizes of all medications to be administered at the time. The patient for which the medication(s) is due is identified by `patientId` and `location`.

Algorithms used to generate the overall schedule must achieve a desirable tradeoff between its quality and the qualities of schedules for individual patients. An overall schedule is unacceptable if it fails to meet any of the hard constraints for the patients. The quality of an acceptable overall schedule is measured by the average time spent by the user (and the dispenser) per dose administered.

## **6 Summary and Future Work**

The primary focus of this paper is on medication dispensers and their integration in the system of tools for minimizing medication errors. Previous sections described the design and operations of a fully automatic personal dispenser, as well as a prescription authoring tool that can translate user's prescriptions and OTC drug directions to a machine-readable specification for such a dispenser. We also briefly described an architecture of automated dispensers designed to take

care multiple patients in a care providing institution. We are building proof-of-concept automated personal dispenser and authoring tool prototypes from components. Some of the components (e.g., MSS and dosage selection module) can also be used as building blocks of professional dispensers, while some others components (e.g., data extraction module of the authoring tool) need to be simplified or eliminated all together for professional use.

Ours is a work in progress. – This statement is true not only for our project specifically but also for developers of medication administration devices in general. – Thus far, our work concentrates primarily on technical aspects. There are many challenges. An example is scheduling: The underlying model for medication scheduling and feasibility analysis [29, 32] of interacting medications differ significantly from real-time models. New algorithms are needed since almost none of the existing algorithms and theories work well. Other challenges include design of dependable dispensing mechanism, good dispenser interfaces, and so on. Nevertheless, these and other technical challenges by and large are surmountable, given time and effort, especially where problems and solutions can be rigorously formulated and evaluated.

Many (and for many reasons challenging) problems arise from the lack of standards. Existing body of open standards (e.g., DICOM) falls short in many aspects. Previous sections already discussed the absence of widely adopted standard document and form schemas for prescriptions and drug directions. Standard schemas can significantly simplify automatic processing of prescriptions. This paper devoted considerable amount of space on medication schedule specification. The key issue here is what should be specified by a specification. While most users cannot digest detailed instructions and complex rules, an automatic dispenser can; in particular, it can take advantage of information to provide scheduling flexibility safely.

## **Acknowledgements**

This work is partially supported by Taiwan Academia Sinica thematic project SISARL. The authors wish to thank members of SISARL medical advisory board, listed at <http://sisarl.org>, and Dr. D. H. Burkhardt for their critiques and suggestions.

## **References**

- [1] Veacez, P. J., “An individual based framework for a study on medical error,” *International Journal for Quality in Health Care*, Vol. 18, No. 4, May 2006.
- [2] “Preventing medication errors,” Report Brief, Institute of Medicine of the National Academies, <http://www.iom.edu/Object.File/Master/35/943/medication%20errors%20new.pdf>,
- [3] Lisby, M., L. P. Nielsen and J. Mainz, “Errors in the medication process: frequency, type, and potential clinical consequences,” *International Journal for Quality in Health Care*, Vol. 17, No. 1, 2005.



- [4] Law, A. V., M. D. Ray, K. K. Knapp, and J. K. Balesh, "Unmet needs in medication use process: perceptions of physician, pharmacists, and patients," *Journal of the American Pharmaceutical Association*, Vol. 43, No 3, 2003.
- [5] Wertheimer, A. I., and T. M. Santella, "Medication compliance research," *Jr. of App. Res. in Clinical and Experimental Therapeutics*, 2003.
- [6] "Pharmacy – nursing shared vision for safe medication use in hospitals: executive session summary," *Am. Journal of Health – Syst Pharma*, Vol. 60, 2003.
- [7] P. Bonnabry, "Information technologies for the prevention of medication errors," *European Pharmacotherapy*, 2003.
- [8] S. C. Dursco, "Technological Advances in Improving Medication Adherence in the Elderly," *Annals of Long-Term Care: Clinical Care and Aging*, Vol. 9, No. 4, 2001.
- [9] Nugent, C. D., D. D. Finlay, N. D. Black, T. Falas, C. Papadopoulus, and G. A. Papadopoulus, "Mobile management and prescription of medication," where
- [10] MEDICATE (Medical Diagnosis, Communication and Analysis Throughout Europe) project, <http://www.chime.ucl.ac.uk/work-areas/ehrs/Mediccate/index.htm>.
- [11] Barlow, J., <http://www3.imperial.ac.uk/innovationstudies/research/healthcare/>, "Health and social care delivery research," 2006.
- [12] CQPI Research Projects, <http://www.engr.wisc.edu/centers/cqpi/projects.html>
- [13] Patient Safety Challenge Grants. AHRQ Publication No. 04-P013, Agency for Healthcare Research and Quality, Rockville, MD, <http://www.ahrq.gov/qual/ptsfchall.htm>.
- [14] A project on identifying and preventing medication errors, Institute of Medicine of National Academies, <http://www.iom.edu/CMS/3809/22526.aspx>.
- [15] Rx Showcase, [http://www.rxinsider.com/prescription\\_dispensing\\_automation.htm](http://www.rxinsider.com/prescription_dispensing_automation.htm)
- [16] e-pill, "Pill Dispenser.": <http://www.epill.com/dispenser.html>.
- [17] Pill boxes: [http://www.dynamic-living.com/automated\\_medication\\_dispenser.htm](http://www.dynamic-living.com/automated_medication_dispenser.htm)
- [18] My Pill Box at <http://www.mypillbox.org/mypillbox.php>
- [19] Murray, M. D., "Automated medication dispensing devices," Chapter 11 in *Making health care safer: a critical analysis of patient safety*, 01-E58, Agent for Healthcare Research and Quality, 2001.
- [20] Luce, A. and A. Leung, "Review of 2005-2006 RERC-AMI National Student Design Competition" <http://www.rerc-ami.org/ami/tech/tr-ami-004-sr-des-med-dispensing/index.aspx>
- [21] PDRHealth, Drug Information, [http://www.pdrhealth.com/drug\\_info/](http://www.pdrhealth.com/drug_info/)
- [22] Health Information Systems: <http://www.hhs.gov/healthit/ahic.html>, and Baron, R. J., E. L. Fabens, M. Schiffman and E. Wolf, "Electronic health records: just around the corner? Or over the cliff?," *Annals of Internal Medicine*, 2005.
- [23] R. W. Park, S. S. Shin, Y. I. Choi, J. O. Ahu, and S. C. Hwang, "CPOE and Electronic Medical

- Record Systems in Korea Teaching Hospitals,” *Journal of American Medical Informatic Association*, September 2005.
- [24] D. M. Cutler, N. E. Feldman, and J. R. Horwitz, “U. S. Adoption of Computerized Physician Order Entry Systems,” *Health Affairs*, Vol. 24, No. 6, 2005.
- [25] R. L. Davis, “Computerized Physician Order Entry Systems: The Coming of Age for Outpatient Medicine,” *PLoS Medicine*, September 2005.
- [26] B. Koppel, *et al.*, “Role of Computerized Physician Order Entry Systems in Facilitating Medication Errors,” *Journal of AMA*, Vol. 293, No. 10, 2005.
- [27] Smaling, J. and M. A. Holt, “Integration and automation transform medication administration safety,” *Health Care Management Technology*, April 2005.
- [28] H. C. Yeh, P. C. Hsiu, C. S. Shih, P. H. Tsai and J. W. S. Liu, “APAMAT: A Prescription Algebra for Medication Authoring Tool,” *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, October 2006.
- [29] P. H. Tsai, H. C. Yeh, C. Y. Yu, P. C. Hsiu, C. S. Shih and J. W. S. Liu (2006), Compliance Enforcement of Temporal and Dosage Constraints, *Proceedings of IEEE Real-Time Systems Symposium*.
- [30] Capsule Technologie, Medical device interface library.
- [31] P. C. Hsiu, H. C. Yeh, P. H. Tsai, C. S. Shih, D. H. Burkhardt, T. W. Kuo, J. W. S. Liu, T. Y. Huang, “A General Model for Medication Scheduling,” Institute of Information Science, Academia Sinica, Taiwan, Technical Report TR-IIS-05-008, July 2005.
- [32] Tsai, P. H. and J. W. S. Liu, “Consistency and feasibility of flexible demand-supply constraints,” Institute of Information Science, Academia Sinica, Taiwan, Technical Report TR-IIS-07-001, January 2007.