Intelligent Tools for Reducing Medication Dispensing and Administration Error

Pei-Hsuan Tsai[1] and Jane W.S. Liu[2]

[1] Inst. of Manufacturing Info. and Sys., National Cheng-Kung University, Tainan, Taiwan
phtsai@mail.ncku.edu.tw
[2] Institute of Information Science, Academia Sinica, Taipei, Taiwan
janeliu@iis.sinica.edu.tw

Abstract. This paper presents an overview of smart medication dispensing and administration devices and software tools designed to minimize dispensing and administration errors. Some of them are for users who take medications on a long term basis without close professional supervision; others are for pharmacy and nursing staffs in hospitals, long term care, and assisted living facilities. These tools should be configurable, customizable, easy to use and effective for diverse users and care-providing institutions. The paper describes approaches taken to meet these objectives.

Keywords: Automated medication management, medication scheduling barcode medication administration, intelligent monitor and alert.

1 Introduction

For years now, professional literature and mass media have been telling us almost everyday of new drugs that can do wonders in curing some previously fatal diseases or help people live with the diseases and chronic conditions better and longer. Unfortunately, they also tell us too often stories (e.g., [1-4]) about medication errors and serious consequences of the errors.

As defined by US FDA (Food and Drug Administration), a medication error is "any preventable event that may cause or lead to inappropriate medication use or patient harm while the medication is in the control of the health care professional, patient, or consumer" [5]. Despite efforts in recent decades to improve medication safety, the rate of medication error remains high even in technologically advanced countries. For example, according to a report published by Institute of Medicine in 2006 [6], a typical (US) hospital patient is subjected to an average of at least one medication error per day. Apparently, this statement is still true today. The Medication Safety Basics webpage of the US CDC (Center for Disease Control and Prevention) [7] states that annually, ADEs (Adverse Drug Events) lead to 700,000 emergency department visits, 120,000 hospitalizations, and $3.5 billion extra medical costs in USA. (Globally, the estimated cost of medication errors is €4.5 – 21.8 billion [8]). CDC expects the numbers of ADEs to grow for reasons including an aging

J. Gibbons and W. MacCaull (Eds.): FHIES 2013, LNCS 8315, pp. 1–21, 2014.

population and increasing use of medications for disease prevention. Indeed, according to [9-11], 40 % of people over 65 take 5-9 medications, and 18% of them take 10 or more, on a long term basis without close professional supervision. A consequence is that the rate of ADEs for elderly individuals is many times that of younger people.

These alarming statistics have motivated numerous efforts in development, deployment and assessment of guidelines, methods, systems and tools for prevention of medication errors (e.g., [12-25]). Medication errors can occur throughout the medication use process of ordering, transcribing, dispensing, and administering. Prior to the advent of computerized physician order entry (CPOE) systems [15], errors introduced during ordering and transcribing account for more than half of all errors. CPOE systems are now widely used in hospitals and clinics in developed countries. Data available to date show that together with clinical decision support and electronic patient health record (ePHR) and medication administration record (eMAR) systems, CPOE has been effective [16], [17], [23] [24]. For example, according to Radley, *et al.* [16], data available in 2008 indicate that electronic prescribing through CPOE systems led to 41-55% reduction of prescription errors in US hospitals. Based on the adoption rate of CPOE systems in USA at the time, they estimate a 12.5% reduction in total medication errors, or approximately 17 million fewer errors per year.

Next to prescribing errors, medication administration errors are the most prevalent and contribute 25 – 40% of all preventable medication errors in hospital settings. An *administration error* is a failure to comply with medication directions due to the administration of a wrong drug, with a wrong dosage, at a wrong time, via a wrong route, or to a wrong patient. The chance of making such mistakes can be reduced when medications and patients are identified by their barcode identifiers and when the right medications are given to the right patient is verified at each administration time with the help of barcode medication administration (BCMA) [18] and eMAR systems. Published data (e.g. [20-22]) have shown that institutions using these systems can reduce administration errors significantly (e.g., by 41%) despite challenges in using them in suboptimal settings and potential errors introduced by workarounds [22]. Today, barcode medication administration and dispensing are supported by modern bedside computers, medication carts and medication dispensing systems such as the ones listed at [26] and [27].

This paper presents an overview of two systems of intelligent tools for the reduction of medication dispensing and administration errors: They are *iMAT* (*intelligent medication administration tools*) and *MeMDAS* (*medication management, dispensing and administration system*). iMAT [28-30] is a family of prototype embedded devices, mobile applications and software tools designed to help their users stay compliant to medication directions while providing them with flexibility in scheduling whenever possible and customization in monitoring and alert capabilities. The targeted users are people who take medications outside of health-care institutions, including elderly individuals and people with chronic conditions living independently.

Fig. 1 shows how iMAT fits in the tool chain for medication use process: An iMAT mobile application or embedded device can be used as a point-of-service tool by a hospital to extend its care of a discharged patient who must remain on a rigorous medication regimen for weeks and months. Such tools may also be consumer electronics purchased by the users themselves or by their friends and family members. Indeed, by searching the web and specialty stores, one can find hundreds of pillboxes, medication managers, etc. (e.g., [31-33]) with subsets of iMAT functionalities.

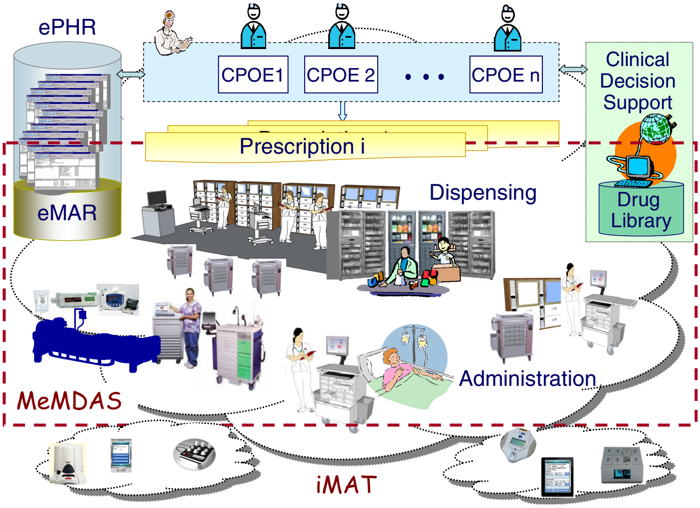


Fig. 1. iMAT and MeMDAS in Tool Chain for Medication Use Process

MeMDAS [34], [35] is a system of intelligent medication dispensing and administration tools for pharmacy and nursing staffs in hospitals, long term care, and assisted living facilities. As Fig. 1 shows, it complements CPOE systems within the tool chain for medication use process. In addition to supporting experimentation with BCMA and barcode medication dispensing (BCMD) by Taiwan University Hospital, the prototype was built to demonstrate several novel concepts and capabilities. They include configurability and customizability of the tools, not only by IT staff of the institutions but also by end-users themselves, and an intelligent monitor, alert and notification (iMAN) tool for detecting user-specified event and action sequences that warrant alerts and notifications be sent to designated person(s) and tools.

The remainder of the paper is organized as follows. Sections 2 and 3 present the motivations, use scenarios and distinguishing capabilities and characteristics of iMAT and MeMDAS, respectively. Section 4 discusses a model-based approach for design, implementation and evaluation of these tools. Section 5 concludes the paper.

2 iMAT - Intelligent Medication Administration Tools

Again, iMAT [28-30] is a system of devices and software tools for reducing the rate and severity of ADEs for people taking medications at home and work, during travels, and so on. Our work on iMAT was a major thrust of the SISARL (Sensor Information Systems for Active Retirees and Assisted Living) Project (2006-2009) [36]. As the project name indicates, its research focus is on technologies for building personal and home automation and assistive devices and systems that can improve the quality of life and self-reliance of their users. Targeted users include the growing population of elderly individuals and people with chronic conditions who are well enough to live active, independent lifestyles. Such a person may take many prescriptions and over the counter (OTC) medications (e.g., more than 5 or 10 according to [9] and [10]) and health supplements for months and years.

2.1 Objectives and Rationales

The benefits of devices and systems that can help users stay compliant and avoid ADEs are self-evident and, no doubt have motivated the vast variety of pillboxes, medication schedulers, etc. [31-33] on the market today. A close look at these tools shows that they typically require the user to load the individual doses of medications into the device, understand their directions and program the device to send reminders accordingly. In other words, they do not address some of the common causes of non-compliance, including misunderstanding of medication directions, inability to adhere to complex medication regimens, and inconvenience of rigid schedules. In contrast, iMAT is designed specifically to remove these causes of noncompliance.

A user of a iMAT medication dispenser and schedule manager has no need to understand the directions of her/his medications. To eliminate the need, iMAT enables the pharmacist of each user to extract a machine-readable *medication schedule specification* (*MSS*) from the user's prescriptions and OTC directions. Once the user's MSS is loaded into his/her iMAT dispenser or schedule manager, the tool automatically generates a medication schedule that meets all the constraints specified by the specification. Based on the schedule, the tool reminds the user each time a dose of some medication should be taken and provides instructions on how the dose should be taken (e.g., with 8 oz of water, no food within 30 minutes, etc.). In this way, iMAT helps to make complex regimens easy to follow.

For users on medications for months and years, tardiness in response to reminders is unavoidable. Directions of modern medications typically provide some flexibility in choices of dose size and time and instructions on what to do in case of late or missed doses. The iMAT scheduler uses scheduling models and algorithms [37-39] that can take advantage of this leeway to make the user's medication schedule easy to adhere. To tolerate user's tardiness, the tool monitors the user's response to reminders, adjusts the medication schedule as instructed by the MSS when the user is tardy, and when a non-compliance event becomes unavoidable, sends alert and notification in ways specified by MSS (e.g., notify the user's physician) and the user (e.g., record the miss). Thus, the tool helps to reduce the rate and ill effects of non-compliance.

Dispensing can be a weak link in medication safety for people taking medications outside care-providing institutions. A typical user targeted by iMAT may be under the care of multiple physicians and given prescriptions ordered via multiple CPOE systems. While each of the user's prescriptions is error free, it may fail to account for interactions between medications ordered by other prescriptions. The user may also take OTC and herbal medicines that may interact with her/his prescription drugs. To reduce the chance for errors due to drug interactions, iMAT imposes on the user two usage restrictions: First, the user lets his/her iMAT manage all his/her prescription and OTC medications since no tool can be effective otherwise. Although the tool does not manage food, it must take into account of user's preferences and habits in meals and snacks when they interfere with some of the user's medications. Second, the user's MSS is generated from the directions of all his/her medications by the user's pharmacist using an authoring tool [30], [40]. A function of the tool is to merge all the human-readable directions and translate them to a standard machine-readable format to enable automatic scheduling as stated above. The other critical function is to check all the directions for possible conflicts (i.e., drug interactions that have not been properly taken into account by the user's prescriptions and directions). When the tool detects possible conflict(s), it alerts the pharmacist to have the conflict resolved. For this purpose, as well as for the generation of MSS, iMAT also has a database containing medication directions in XML format, the format used by the iMAT prototype. We will return shortly to provide specifics. Fig. 2 shows where this tool, called the MSS Authoring tool in the figure, is used in a scenario assumed by iMAT.

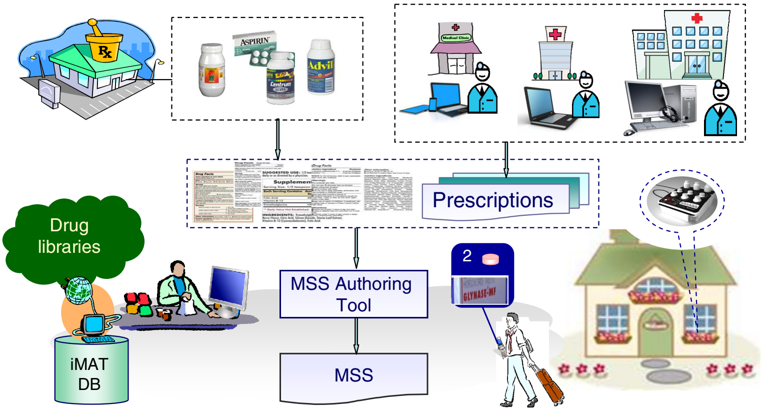


Fig. 2. iMAT Use scenario (From [30])

2.2 Structures and Key Components

Specifically, Fig. 2 assumes that the user has a medication dispenser. It holds his/her medications and helps him/her take them at home: The tool makes sure that the user retrieves the right dose of each medication from the right container when he/she responds to reminder at each dose time. The medication scheduler component of the dispenser can serve as a schedule manager. When the user is away from home and carries the medications with him/her on the road, the schedule manager provides reminders to the user by sending text and voice messages, with or without pictures of the medications, to his/her smart phone, as the lower right part of the figure illustrates. When the dispenser is connected to the Internet, the user can acknowledge the receipt of each reminder and report his/her action taken as response to the reminder.

The scenario assumes that the user's pharmacy supports the dispenser and has access to the user's medication record. When the user goes to fill a new prescription or purchase some OTC drug, the pharmacist uses the MSS authoring tool to process the new direction(s) together with the directions of medications currently taken by the user according to the user's record and after making sure all conflicts have been resolved, generate a new MSS for the user's dispenser. The pharmacist gives the user the MSS in a memory card (or loads the MSS to the user's dispenser via the Internet) and new supplies of medications in containers. Each container holds a medication marked by the universal identifier of the medication.

Fig. 3 shows two dispenser prototypes. They are similar in general: Each dispenser has a base. Medication containers are plugged in the base. An interlock mechanism in the base makes sure that the containers, once plugged in, are locked in place. A container can be removed, or its lid can be opened, only when a dose of the medication in it is due to be taken and the user responds in time to the dispenser's reminder by pressing the PTD (Push-To-Dispend) button. The prototype on the left tags each container by the radio frequency identifier (RFID) of the medication. When the user plugs a container into an empty socket in the base, the action triggers the dispenser controller to read the tag of the container and thus identify the medication located at that socket. The prototype on the right identifies the medications in the containers by their barcode ids. The one shown here has a window in the front with a barcode scanner and camera behind it. The user scans the barcode of each new container and then plugs the container into an empty socket and thus enables the dispenser to recognize and locate the medication. Both prototypes can use a built-in webcam to make sure that the dose sizes retrieved by the user are correct.

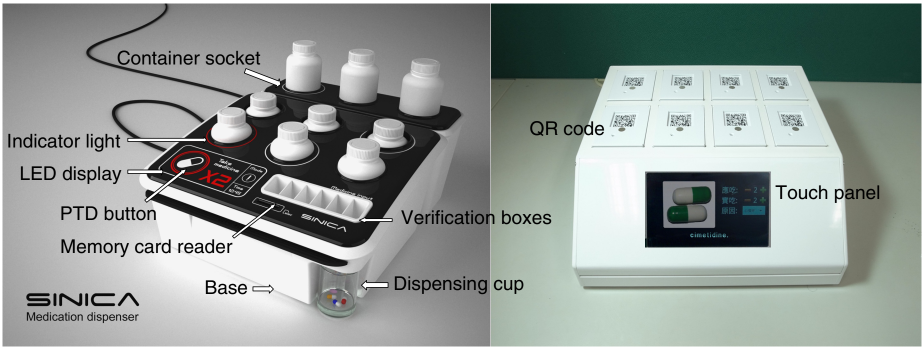


Fig. 3. Physical appearance of two prototype dispensers

Fig. 4 shows alternative configurations of iMAT: The dotted box encircles the software components and data that are essential for all configurations. A user who does not want a stand-alone dispenser can have these components run on a smartphone platform, as a self-contained schedule manager and monitor such as the one described in [41]. The mobile application offers most of the schedule management and compliance monitoring functions of an iMAT dispenser. This configuration is depicted in the right part of the figure.

The flexible configuration shown in the left half of the figure may be chosen by a user who has multiple computers and mobile devices at home and work: The software components run on a PC and uses one or more laptop computers and mobile devices for its interaction with the user. A user may start with only these parts and incur no expense of special-purpose hardware. As he/she starts to take more and more medications, the user can get one or more dispensers, less the software components, and connect them to the computer as peripheral devices. As an example, Fig. 4 shows a dispenser connected locally to the PC and a dispenser connected remotely for a user who has a dispenser at home and another at work.

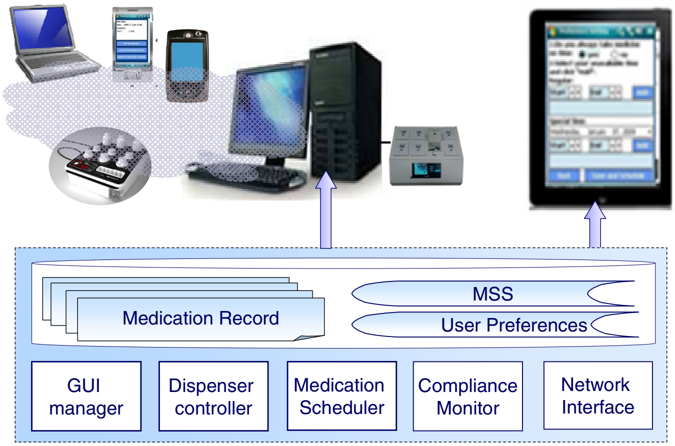


Fig. 4. Alternative configurations (From [30])

2.3 Medication Schedule Specification

As stated earlier, the prototype iMAT intends to demonstrate the feasibility and safety of flexible medication schedules. Today even low-cost computing devices can store gigabytes of data and carry out mega-instructions per second. A medication schedule manager can take advantage of such resources to exploit fully timing and dosage flexibilities provided by directions of most medications and thus make the medication schedules less rigid, more considerate of user preferences, and more tolerant to user tardiness. The scheduling models and several heuristic algorithms for scheduling multiple medications for this purpose can be found in [28] and [37-39].

Firm and Hard Constraints. The iMAT medication scheduling model incorporates the concept of firm and hard constraints that has been developed and used to build safety critical real-time systems [42]. The remainder of this section describes how these constraints are defined within the medication schedule specifications.

We use the terms firm and hard constraints in the same sense as they are used in real-time systems literature. *Firm constraints* are typically more stringent. The scheduler tries to meet all the firm constraints whenever possible. Violations of firm constraints can occur, often due to user's tardiness or forgetfulness. The scheduler allows such violations when it cannot find a schedule to meet all firm constraints.

When the user is so late that a firm timing constraint is violated, the schedule manager recomputed the time for the late dose and possibly a new dose size as specified by the MSS. Such adjustments in the schedule may degrade the rigor of compliance (i.e., increase the change of ADE) and quality of the schedule (e.g., be less convenient for the user) but are nevertheless acceptable. Take Fosamax as an example. According to PDRHealth [17], Fosamax in tablet form is used to prevent or treat oosteoporosis. Suppose that the user is directed to take a 5 mg tablet every morning at least 30 minutes before taking any other medication or food. The direction also says that in case of a miss dose, skip the missed dose and resume the regular schedule the next day. So, a tablet each day is a firm constraint. The MSS may simply say to cancel the current dose and inform the user of the cancellation if he/she responds to the reminder for taking Fosamax too late: Here, being too late means after eating breakfast or when some other medication must be taken without delay.

Hard constraints are less stringent; they limit the degree to which medication directions are allowed to be relaxed and schedule quality to degrade. A violation of a hard constraint is treated as a non-compliant event and warrants an action (e.g., warn the user, call a designated family member, alert the user's doctor, and so on.) Clearly, the action depends on the medication, the user, and the severity of the violation; it is specified by the MSS. In case of Fosamax, the user's MSS may say to treat 7 consecutive missed doses as a non-compliance event. So, a hard constraint is time between consecutive dose is no greater than 7 days. When the event occurs, the user (or a family member) is alerted. He/she may ask his/her physician to change the prescription to "one 35mg tablet once a week". By switching to a less frequent schedule, he/she risks a higher chance of side effects such as painful acid reflux.

In general, for each medication *M* managed by the user's iMAT, the MSS contains a section extracted by the authoring tool from an XML file of directions stored in the iMAT database. Table 1 lists the parameters contained in the section for *M*.

Specifically, the section of *M* in MSS provides general information (e.g., name(s), granularity and picture(s) of the medication and the duration the user is supposed to be on it). The tool needs this information to manage and schedule the medication. The section has a dosage parameters (DP) part: DP defines the size and timing constraints for doses of *M* when the medication does not interact with other medications of the user. If some of the user's medications interact with *M*, the section also contains a special instructions (SI) part. SI specifies changes in dosage parameters and additional timing constraints to account for the interactions.

Dosage Parameters. The parameters in lines 1 and 2 in the DP part of the table define firm constraints of the medication. Specifically, line 1 gives the *nominal dose size range*; it bounds the sizes, in term of multiples of granularity of *M*, of individual doses of *M*. Line 2 gives the *nominal separation range* in terms of the minimum length and maximum length of time between two consecutive doses of the medication. The medication scheduler computes the normal schedule of the medication based on these parameters.

Table 1. Parameters of medication schedule specification (From [30])

- M: Name of the medication
- g: Granularity of dose size
- $[T_{min}, T_{max}]$: Minimum and maximum durations
- Other relevant attributes
- Dosage Parameters (DP)
  1. $[d_{min}, d_{max}]$: Nominal dose size range
  2. $[s_{min}, s_{max}]$: Nominal separation range
  3. $[D_{min}, D_{max}]$: Absolute dose size range
  4. $[S_{min}, S_{max}]$: Absolute separation range
  5. (B, R): Maximum intake rate defined by an upper bound B of total size of doses in a specified time interval R
  6. (L, P): Minimum intake rate defined by a lower bound L of total size of doses in a specified time interval P
  7. Non-compliance event types and corresponding actions.
- Special Instructions (SI)
  1. N: Name of an interferer
     a. Change list
     b. $\sigma_{min}(M, N)$: Minimum separation from M to N
     c. $\sigma_{min}(N, M)$: Minimum separation from N to M
  2. L: Name of another interferer

     ▪ ▪ ▪

In contrast, the *absolute dose size* range and *absolute separation range* in lines 3 and 4, respectively, define hard constraints. The medication scheduler never uses dose size and separations outside these ranges. The constraints in lines 5 and 6 are called *maximum intake rate* (B, R) and *minimum intake rate* (L, P), respectively. The former specifies the total size of all doses within any time interval of length R to be no more than B. The latter requires that total size of all doses within any interval of length P to be at least equal to L. The scheduler treats these constraints (or the less stringent rates $(B+\beta, R)$ and $(L- \lambda, P)$ for some small β and λ no less than zero) as hard constraints.

As an example, the direction of Tylenol reads “Take one tablet every 4 to 6 hours. If pain does not respond to one tablet, two tablets may be used. Do not exceed 8 tablets in 24 hours.” The DP part of this medication has $[d_{min}, d_{max}] = [1, 2]$, $[s_{min}, s_{max}] = [4, 6]$, $(B, R) = (8, 24)$; granularity of time is one hour. The values of these parameters follow literally from the direction. Since the drug is to be taken as needed, $[D_{min}, D_{max}] = [0, 2]$ and $[S_{min}, S_{max}] = [4, \infty]$. Moreover, there is no required minimum total dose size for this drug; hence $(L, P) = (0, 24)$.

While the maximum intake rate is imposed to prevent overdose, the minimum intake rate constraints the number of missed doses. As an example, suppose that the physician of a user taking Propranolol for hypertension ordered for him/her one low dose tablet 3 times a day and wants to make sure that he/she does not skip any dose, or at most a dose occasionally. This constraint is specified as $(L, P) = (3, 24)$, or more relaxed $(L, P) = (2, 24)$ or $(20, 168)$ (i.e., skip one dose per day or one per week).

Finally, we note that the “if you miss a dose” instruction within directions typically leads to an absolute separation range $[S_{min}, S_{max}]$ containing the nominal range. As an example, the nominal and absolute separation ranges of a once a day medication are [24, 24] and [12, 48] or [8, 48], respectively, when its missed dose instruction reads "If you miss a dose, take it is when you remember. If it is close to the time for the next dose, skip the one you miss and go back to regular schedule."

Special Instructions (SI). In Table 1, the term an *interferer* of a medication *M* refers to a medication (or food) that interacts with *M* so much that some changes in the directions of *M* are warranted. The SI part in the section of MSS for *M* has an entry for each interferer *N* of *M.* The dose size and separation ranges of *M* may need to be changed to take account of their interactions. Such changes are specified by the change list in the entry. The dosage parameters in the change list are in effect as long as the user is on both *M* and *N*.

The entry for an interferer *N* may also define additional separation constraints: The time separation between each dose of *M* and any dose of the interferer *N* must be within the specified range: The minimum separation σ_{min} (*M, N*) from *M* to *N* specifies a lower bound to the length of time from each dose *M* to any later dose of *N*, and σ_{min} (*N, M*) from *N* to *M* is a lower bound to the time from each dose to *N* to any later dose of *M*. For example, the constraint that Fosamax must be taken before any food and at least 30 minutes before breakfast is specified by σ_{min} (Fosamax, Food) = 30 minutes and σ_{min} (Food, Fosamax) = 4 hours. The technical report [39] discusses additional constraints, such as maximum inter-medication separation constraints.

3 Medication Management, Dispensing, and Administration

Again, the acronym MeMDAS stands for Medication Management, Dispensing and Administration System [34], [35]. It is a distributed system of tools that supports medication dispensing and administration stages of medication use process as shown in Fig. 1. Its primary users are nursing and pharmacy staffs in hospitals, long term care, and assisted living facilities.

3.1 Capabilities

The system provides the users with tools similar to the ones in state-of-the-art mobile carts and medications stations (e.g., [26] [27]), as well some distinct capabilities:

- Medication (and medical supply) delivery and inventory monitoring;
- Barcode medication dispensing and administration (BCMD and BCMA);
- Work and time management (WTM);
- Configuration and customization tools and user interface functions;
- Information access and labor-saving capabilities, such as generating shift report from data and notes collected during the user's shift, tracking medication and medical supply usages and automating requests for replenishments; and
- Intelligent monitor, alert and notification (iMAN) [42].

A user can use the modern WTM tool as a personal digital assistant and have it maintain not only medication schedules and appointments of patients under the user's care, but also track the user's daily work plans and personal schedule (e.g., meetings and tasks). For example, it enables the user to schedule times for various tasks (e.g., prepare a patient three days ahead of the patient's colonoscopy appointment and warm up a patient before a physical therapy session) and gets reminders from the tool at those times. Adjustment in patient schedules is unavoidable. The WTM tool is capable of enforcing rules governing patient schedules. A change of the schedule time of a patient event requested by the user can take effect only after the tool has confirmed that the changed schedule satisfies all rules.

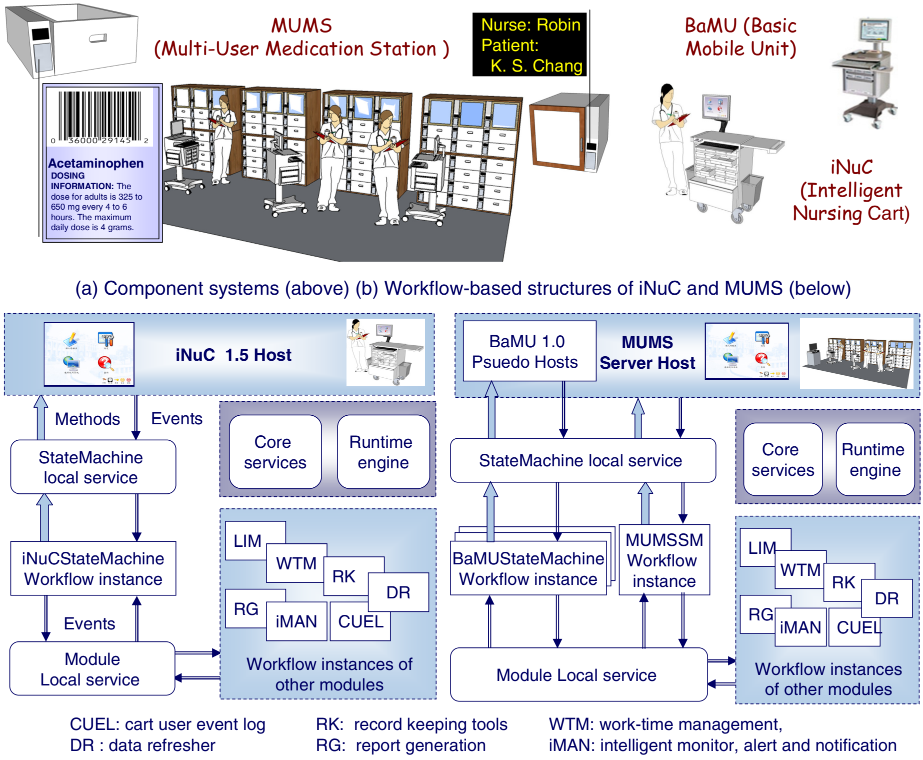
Just like iMAT described in the previous section, flexibility is a distinct characteristic of MeMDAS. By being *flexible*, we mean easily configurable and customizable. Medication administration processes vary from hospital to hospital, department to department, and even patient to patient. Protocols and rules governing ideal administration processes for a patient in an ICU (Intensive Care Unit) and a patient in a general ward are typically different. For this reason, MeMDAS tools and component systems are built to be easily configurable and customizable, in most cases by end users themselves: Nursing and pharmacy staffs with proper authorization can customize for themselves majority of the MeMDAS tools and user interface functions to follow the protocols and enforce the policies and rules of their respective institutions, departments and patient wards.

The intelligent monitoring and notification tool [42] complements interlock and control mechanisms to enhance error prevention. Like similar tools for safety critical systems, iMAN also enables the user to analyze and determine the causes of errors after they occur. iMAN is unique in its capability to detect events and action sequences deemed by the user as having a high likelihood to cause errors and when such an event is detected, notify designated persons to take preventive actions. An easy to use and reliable iMAN is particularly important for monitoring and tracking common workarounds and protocol violations such as the ones reported in [22].

3.2 Component Systems

A MeMDAS has three types of component systems. They are MUMS (multiple-user medication station), iNuC (intelligent nursing carts), and BaMU (basic mobile unit) as depicted in part (a) of Fig. 5.

The term *medication station* refers to a system of smart cabinets with barcode controlled containers monitored and operated by a small server. Medication stations such as the ones listed in [26] and [27] typically operate in fully automated mode: When a user comes to retrieve medications for a patient, the station opens automatically all the containers holding the medications due to be administered to the patient at the time. Operating in this mode, a station can serve only one user at a time. In a ward with many nurses (e.g., 5-10) caring for patients on frequent medications, the added burden on the nurses to stand in line for retrieval of medications or to adjust their work plans in order to minimize queuing time often more than offsets the advantages of using the station. MeMDAS medication stations are configurable so that they can also operate in a semi-automatic mode. When operating semi-automatically, the station server collaborates with the users and their mobile carts to ensure correct dispensing of medications to multiple users concurrently.


MUMS
(Multi-User Medication Station)
Nurse: Robin
Patient:
K. S. Chang
BaMU (Basic
Mobile Unit)
iNuC
(Intelligent
Nursing Cart)
0 36000 29145 2
Acetaminophen
DOSING
INFORMATION: The dose for adults is 325 to 650 mg every 4 to 6 hours. The maximum daily dose is 4 grams.
iNuC 1.5 Host
Methods
Events
StateMachine local service
Core services
Runtime engine
iNuCStateMachine Workflow instance
Events
Module Local service
LIM
WTM
RK
DR
RG
iMAN
CUEL
Workflow instances of other modules
BaMU 1.0 Psuedo Hosts
MUMS Server Host
StateMachine local service
Core services
Runtime engine
BaMUStateMachine Workflow instance
MUMSSM Workflow instance
Module Local service
LIM
WTM
RK
DR
RG
iMAN
CUEL
Workflow instances of other modules
CUEL: cart user event log
DR : data refresher
RK: record keeping tools
RG: report generation
WTM: work-time management,
iMAN: intelligent monitor, alert and notification

(a) Component systems (above) (b) Workflow-based structures of iNuC and MUMS (below)

Fig. 5. Component Systems of MeMDAS

Both iNuC and BaMU are mobile nursing carts. iNuC [35] is a self-contained system. Each cart serves a user (usually a nurse). The drawers in it carry the daily doses of medications for each of the patients cared for by the user. The block diagram in the left half of part (b) of the figure shows the structure of a workflow-based version of iNuC. Except for dispensing, an iNuC offers its user all the capabilities listed above without help from MUMS, and, in events of network and hospital information system outages, can function stand-alone.

A BaMU is a light-weight system of mobile tools for use during barcode dispensing of medications from MUMS and for transporting the currents doses retrieved for each patient from MUMS to the patient's bedside. It relies on a MUMS server to provide work planning, scheduling and monitoring and alert functions. Some BaMU do not have the medication administration and patient record keeping tools. Some of such BaMU's are used in wards that have computers at patients' bedsides for these purposes. Such a BaMU can also function as an intelligent medication supply cart for use by pharmacy staff.

3.3 Alternative Configurations

The MeMDAS prototype was partially funded by National Taiwan University Hospital (NTUH) and designed and developed by the SISARL project [36] in close collaboration with the hospital's nursing and pharmacy departments. At the time (2008-2010), the hospital used computers on mobile carts to provide nursing staff with a web-based interface via which they can access the hospital information system and read and update their patients' records. The hospital was in the process of introducing barcode ids of medications and patients. An objective of MeMDAS was to support the hospital's experimentation with BCMA.

The hospital was also planning to experiment with alternative dispensing workflow processes. As with most hospitals in Asian-Pacific region, a centralized dispensing process was used throughout the hospital at the time: According to this process, the pharmacy prepares and delivers daily to each ward a supply cart with drawers. Daily doses of medications for each patient in the ward are in one or more drawers. To support this process, one or more MUMS can be used in the pharmacy, and BaMU's can be used as intelligent supply carts. Together, they support barcode medication dispensing and make the process of preparing supply carts less error prone.

The hospital was planning to adopt distributed dispensing in departments where patients' prescriptions change frequently, including ICU and OR. MUMS and BaMU were intended for such departments: The pharmacy monitors and stocks the cabinets in the station with all or most medications needed for patients in each ward. At times when some medications are due to be administrated to one or more of her/his patients, each nurse retrieves individual doses of the medications for each patient from the cabinets under the control of the station server and his/her BaMU. We will return in Section 4 to describe this collaborative process.

Distributed dispensing tends to increase workload for nurses. Hybrid workflow process is a compromise: In this case, some medications are dispensed and delivered via supply carts by the pharmacy. The ward also has a medication station and uses it to hold controlled drugs and frequently used medications, making it possible for nurses to get newly ordered medications on a timely basis.

In wards where dispensing is centralized or hybrid, nurses uses iNuC for BCMA: To put a medication drawer of a patient under the control of an iNuC, the nurse removes the drawer from the supply cart, scans the barcode patient id in the drawer to capture the id and then puts the drawer in any empty drawer slot of his/her iNuC. Sensing that a drawer is placed in the slot, the RFID reader of the cart reads the tags on the drawers and acquires the association between the id of the new drawer, its location in the cart and patient's barcode id. From this information, it creates the mapping between the drawer location and the patient id. Later, when the patient is due to take some medication(s), the nurse can have the cart open the patient's drawer at bedside by scanning the patient's barcode id in the wristband worn by the patient.

As stated earlier, a distinguishing characteristic of MeMDAS is flexibility: MUMS can be easily configured to run in automatic or semiautomatic mode. The software system controlling operations, GUI and user interactions of the mobile carts can be configured to make an iNuC work as a BaMU and vice versus. This is accomplished by building the component systems on workflow-based structures depicted in part (b) of Fig. 5. Section 4 will elaborate this design choice.

3.4 User-Centric Design and Development

MeMDAS was prototyped in a user-centric manner collaboratively by researchers from the SISARL project and representative users from NTUH. The approach was motivated by studies such as [24] that recommend close involvement of the targeted users throughout the development process as a way to eliminate sources of errors known to occur when new automation systems are deployed.

Following this approach, the requirement capture process for each component system started with presentations of the concepts and functions of the system to representatives of NTUH nursing staffs from many departments and pharmacy staffs. Discussions during these meetings provided the design team with clearer understanding of users' needs, wishes and views. After the meetings, likely early adopters were identified. The development team met with them regularly until concrete use scenarios were developed and requirements were defined and prioritized. Later, when the alpha version of the prototype became available, they used it on a trial basis and their assessments helped to improve the system.

Requirement specifications in textual and diagrammatic forms, even when augmented by formal specifications, are ineffective communication channels between users and developers. For this reason, we used mockups for requirement solicitation, definition and documentation purposes. Take iNuC as an example. Except for the absence of the physical medication drawers with the interlock mechanism and real patient data and nurse ids, the mockup gives the evaluators the look and feel of a real iNuC. We used it to collect information on what the users want the system to do, how the tools interact with them, what and how the display shows, what their preferences in input/output devices and mode of operations are, and so on.

Mockups proved to be as effective as we had hoped them to be. Through them, we indeed obtained valuable feedback and suggestions. They also enabled us to identify design defects and potential bugs that were likely to remain unnoticed until the prototypes are deployed. As an example, during the administration of a medication, iNuC GUI provides a table entry in which the user is required to record the actual quantity of the medication consumed at the time if the quantity differs from the prescribed quantity. This information is used by the tool to track the supplies more accurately. The label and placement of the entry displayed by the iNuC mockup misled some evaluators to think that the user can change the dose size of the medication. Changing dose size by the nurse is, of course, not allowed for most medications. As another example, a sign-in/login feature provided by the mockup for sake of user convenience may leave some of patients unattended during shift changes. To fix this design error, a new role-based access control policy was developed to take into account delays and other anomalies during shift changes.

4 Model-Based Design, Implementation and Evaluation

Flexibility (i.e., configurability and customizability) of MeMDAS prototypes was achieved by building them on a workflow-based architecture. The basic building blocks of a workflow-based application/system are called *activities*. An activity may be the execution of a program, scan of a barcode, transmission of a message, etc. Activities are composed into module-level components called *workflows*. The order and conditions under which activities in a workflow are executed and the resources needed for their execution are defined by the developer of the workflow. The workflow approach [45-49] has been widely used in enterprise computing systems for automation of business processes.

We use the workflow paradigm in two ways: for flexible integration of reusable components and for modeling the system, users and their interactions. Specifically, workflow-based MeMDAS component systems run on Windows Embedded Standard and .NET Workflow Framework [49]. Part (b) of Fig. 5 shows the workflow-based structures of iNuC and MUMS server. The block diagrams intend to highlight the commonalities between iNuC and MUMS Server. (To save space, we omitted the diagram for BaMU, which is essentially the same as that of iNuC.) Only boxes representing the host and state machine workflow are labeled by the names of the systems; the systems differ primarily in these parts. In particular, the state machine local service interface and module local service interface in all systems are the same. Similarly, the workflows provided by other modules are identical. By replacing iNuC state machine workflow with a BaMU state machine workflow, we can make the GUI and the cart behave like a BaMU.

We also use workflows as models of the systems and tools throughout the development process. According to the workflow-based model [50], jobs and tasks done by the system are modeled as activities and workflows. They are called *device activities and workflows*. Many operations of MeMDAS are semi-automatic. Actions taken and tasks done by human user(s) are modeled by *user activities* and *workflows*. Interactions between the user and the system are modeled by local services for workflow to workflow communication. The model also incorporates workflow definitions of GOMS (Goals, Objectives, Methods and Selections) and MHP (Model Human Processor) [51] [52] model elements commonly used in user-interface design to characterize human users with different attributes and skills.

Throughout the development process, the behavior of the new system or tool is defined by its *operational specification*. The specification consists of a model of the system defined in terms of device workflows and a model of the user(s) defined in terms of user workflows. The definitions also specify the resources (e.g., barcode scanner, interlocks, executables, and human users) required by each activity. In the design phase before the actual resources are available, the workflow definitions call for virtual resources (i.e., device simulators, dummy code, user models, etc.) as resource components. As the development process progresses, the virtual resources are replaced by physical devices and programs. Device workflows modeling module-level components become implementations of the components. User workflows provide use scenarios and scripts for testing purposes.

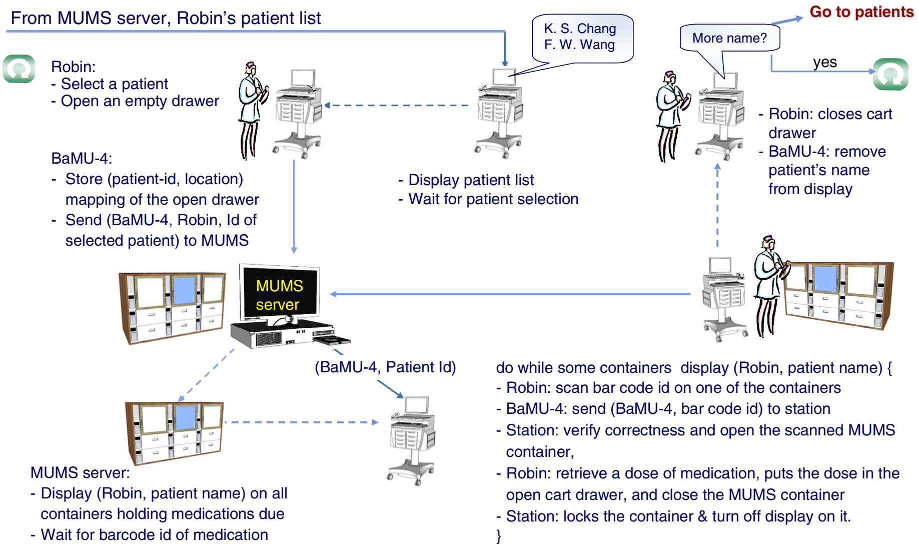
An advantage of specifying operations of the system and modeling user(s) and user-device interactions in terms of workflows is that such specifications and models are executable. By executing the models, the developer can assess the design, usability and performance of the system at each stage of the development process. The simulation environment described in [50] was developed for this purpose.

As a case study, we used workflow models of MUMS, BaMU, and users in a series of simulation experiments to determine whether the MUMS server, multiple users and their BaMUs work correctly in semi-automatic medications dispensing. By being correct, we mean that every user gets correct medications from MUMS and puts the medications in the correct patient drawer for every patient.

Fig. 6 shows a scenario to illustrate the process. The scenario takes place in a ward where a MUMS is used to support distributed dispensing. At the start of each shift, the MUMS server plans for each nurse in service an administration schedule for all of his/her patients. At the time when one or more of his/her patients is due to take some medications, the server sends a reminder to the nurse. In response to the reminder, the nurse (called Robin in Fig. 6) logs on a BaMU (BaMU-4 in the figure) and thus acquires exclusive use of the cart and informs the MUMS server that she is using the cart. She then selects RetrieveMedications command via the GUI of BaMU-4. Upon receiving the command, MUMU server sends the list of Robin’s patients due to take medications at the time to the cart.

The part of the scenario illustrated by Fig. 6 starts from the solid arrow in the upper left corner of the figure. The arrow represents the transmission of patient list. (Other solid arrows also indicate transmission of data while dashed lines or arrows represent elapse time.) Upon receiving Robin’s patient list, BaMU-4 displays the list, unlocks the empty drawers in the cart and then waits for Robin’s selection of a patient from the list. At the MUMS with the cart, Robin selects a patient from the list displayed by the cart and opens an unlocked drawer. Sensing a drawer is opened, BaMU-4 stores the mappings of drawer-location-patient-id and drawer-id-patient-id and sends the mappings to the MUMS server. The server will need them later to ensure that the nurse takes the right drawer to the right patient.

In response to Robin’s patient selection, BaMU-4 sends (BaMU-4, Robin, Patient Id) to the server, informing the server that Robin is retrieving medications for the specified patient. The server responds by having the displays on the containers holding medications to be retrieved for the patient to show Robin’s name and the patient’s name. To retrieve a dose from one of these containers, Robin uses the barcode scanner on BaMU-4 to read the barcode on the label of the container. The captured reading is sent to the MUMS server. After verification, the server unlocks the container, allowing Robin to retrieve a dose from it. Once Robin finishes retrieving all the medications of the selected patient, she closes the open cart drawer.


From MUMS server, Robin's patient list
K. S. Chang
F. W. Wang
More name?
Go to patients
yes
Robin:
- Select a patient
- Open an empty drawer
BaMU-4:
- Store (patient-id, location) mapping of the open drawer
- Send (BaMU-4, Robin, Id of selected patient) to MUMS
- Display patient list
- Wait for patient selection
- Robin: closes cart drawer
- BaMU-4: remove patient's name from display
MUMS server
(BaMU-4, Patient Id)
MUMS server:
- Display (Robin, patient name) on all containers holding medications due
- Wait for barcode id of medication

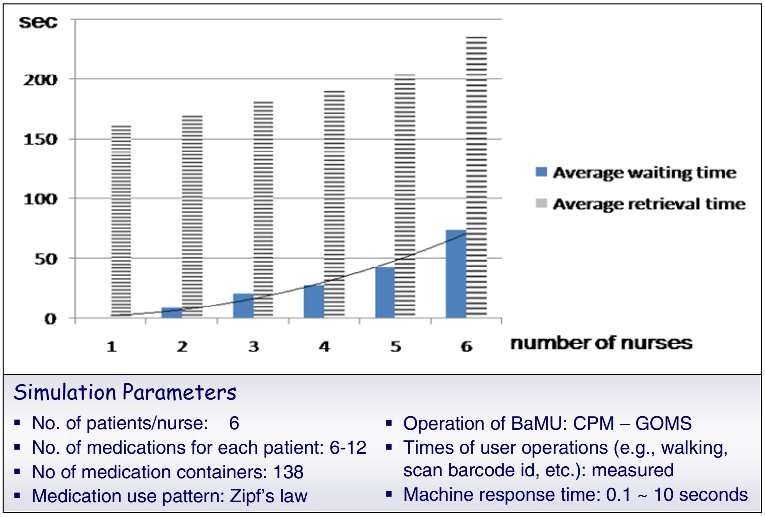
```
do while some containers  display (Robin, patient name) {
- Robin: scan bar code id on one of the containers
- BaMU-4: send (BaMU-4, bar code id) to station
- Station: verify correctness and open the scanned MUMS
  container,
- Robin: retrieve a dose of medication, puts the dose in the
  open cart drawer, and close the MUMS container
- Station: locks the container & turn off display on it.
}
```

Fig. 6. Semi-automatic dispensing from MUMS

The drawer is locked in place in the cart. Robin can have the drawer opened and removed only by scanning the barcode id of the patient.

In all the simulation runs carried out in our attempt to detect errors and malfunctions caused by user-system interactions, we observed none. Examination of event logs generated during simulation shows that the complicated process illustrated by Fig. 6 in fact is not error prone: While the MUMS server allows multiple users to access the cabinets at the same time, it correctly permits only one user at a time to open any container. The fact that each user has exclusive use of his/her own cart and barcode scanner is the main reason.

We also simulated different numbers of users retrieving medications from a MUMS to determine the responsiveness the system as a function of concurrent users. The estimated times taken by a user to operate the BaMU GUI were obtained using the CPM variant of the GOMS model [51]. The amounts of time for other user actions, (e.g., open a drawer, walk a distance, etc.) were obtained by measuring the amounts of time taken by several test subjects to carry out the actions. Fig. 7 shows the result of such a simulation experiment. In this case, the MUMS has only one cabinet with 138 containers. The result show that the average time spent by a user waiting to access containers remains small compared to the average amount of time required to retrieve medications from containers when the number of nurses using the MUMS concurrently is no more than three.



Simulation Parameters

- No. of patients/nurse: 6
- No. of medications for each patient: 6-12
- No of medication containers: 138
- Medication use pattern: Zipf's law
- Operation of BaMU: CPM – GOMS
- Times of user operations (e.g., walking, scan barcode id, etc.): measured
- Machine response time: 0.1 ~ 10 seconds

Fig. 7. Average retrieval time and waiting time of dispensing from MUMS (From [50])

5 Summary

Previous sections described two systems of tools for reduction of medication administration and dispensing errors. iMAT is for people who take medications on their own, i.e., outside care-providing institutions. MeMDAS provides mobile carts and smart cabinets similar to the ones listed in [26] and [27].

A distinguishing characteristic of both systems is flexibility. iMAT exploits flexibilities provided by directions of typical medications to make the user's medication schedule easy to adhere to and tolerant of user's tardiness whenever scheduling flexibility does not jeopardize compliance. MeMDAS allows users with authorization to tailor the tools for their own institutions, departments and patient wards. Both systems try to prevent errors by providing their users with reminders and instructions on when and how medications are administrated. Both systems enable user actions and anomalous events to be monitored and alerts and notifications to be sent as specified, in case of iMAT, by the user's MSS, and in case of MeMDAS, by users with proper authorization.

Several prototype iMAT dispensers and schedule managers, as well as a MSS authoring tool, were built to prove concepts ranging from medication scheduling, user-centric design, device models and architectures and adaptability. To assess usability and effectiveness of mobile schedule managers, we conducted a small field trial on ten test subjects with ages ranging from 25 to 60 and different education background [28]. Their occupations include student, engineer, housewife, retiree, and businessman. Their medications were for diabetes, hypertension, heart disease, asthma, and other chronic diseases. Before the subjects started to use a version of the iMAT schedule manager running on smart phones, we logged their medication compliance using questionnaires on paper. According to the logs, they missed 1/3 of the doses of their prescription medications and were late in dose time by one to three hours on the average. The results show that the tool was effective (and easy to use) for subjects who are busy and who are often not sure when and what medications should be taken. It was particularly effective in improving compliance for three subjects: They are (1) a 30-year-old software engineer who is a diabetic and takes 11 types of medications at least four times a day; (2) a 24-year-old graduate student who has two medications that must be taken every 12 hours for controlling her asthma; and (3) a 60-year-old businessman who takes eight types of medications at least three times a day for treatment of hypertension and diabetes. Commonalities among them are their complex medication regimens, long work days and irregular hours, and the inconvenience of taking medications during work and at night. As expected, for subjects who do not want to take medications because of other factors, the tool did not work well. Some subjects live at relatively slow pace. They can manage their schedules without the tool and hence were not motivated to use the tool.

A missing piece in the iMAT tool set is a sufficiently complete database of medication directions in the XML format. Currently, the iMAT database contains directions for a few hundred commonly used medications. They were obtained by manual translation of directions in PDRHealth [17] to XML format. We are developing a translator to automate the translation process.

MeMDAS component system prototypes are in different stages of maturity. Their code can be found at the open source software repository http://openfoundry.org and are released under GPL license. A commercial version of the medication station based on the MeMDAS prototype is currently being used on trial basis in a hospital in Taipei, Taiwan.

Acknowledgments This work was partially supported by the Taiwan Academia Sinica thematic project SISARL, a grant from National Taiwan University Hospital, the Flexible Medication Management Systems Project supported by the Taiwan Ministry of Economic Affairs, and by the Center of IET, National Taiwan University under MOE Grant AE-00-00-06.

References

1. Strategies to reduce medication errors: working to improve medication safety, http://www.fda.gov/Drugs/ResourcesForYou/Consumers/ucm143553.htm
2. Medication errors led to severe harm or death in 36 Ontario cases. Canadian Press (August 2013), http://www.cbc.ca/news/health/story/2013/08/21/medication-errors-ontario-hospitals.html
3. Emily's Story, http://emilyjerryfoundation.org/emilys-story/
4. Kowalczyk, L.: Brigham and Women's airing medical mistakes – hospital reports errors to staff in drive for improvement. Boston Globe (April 2013)
5. Medication Errors, Food and Drug Administration, http://www.fda.gov/drugs/drugsafety/medicationerrors/default.htm
6. Preventing medication errors, Institute of Medicine Report. National Academies Press, Washington, DC (2006)
7. US CDC Medication Safety Program, Medication Safety Basics, http://www.cdc.gov/medicationsafety/basics.html#Key
8. Medication errors, European Medicine Agency, http://www.ema.europa.eu/ema/index.jsp?curl=pages/special_topics/general/general_content_000570.jsp

9. Koper, D., et al.: Frequency of medication errors in primary care patients with polypharmacy. Family Practice (June 2013)
10. Budnitz, D.S., et al.: Emergency hospitalizations for adverse drug events in older Americans. New England Journal Medicine (November 2011)
11. Hall, H.: Reducing the risk of adverse drug events. posted in January 2012 at, http://www.sciencebasedmedicine.org/reducing-the-risk-of-adverse-drug-events/
12. Health information technology in the United States: better information systems for better care (2013) Published by Harvard School of Public Health, Mathematica Policy Research and Robert Wood Johnson Foundation, http://www.rwjf.org/content/dam/farm/reports/reports/2013/rwjf406758
13. Addressing medication errors in hospitals: a practical toolkit, California Health Foundation, http://www.chcf.org/publications/2001/07/addressing-medication-errors-in-hospitals-a-practical-toolkit
14. Prevention of medication error, American Academy of Orthopaedic Surgery (2008), http://www.aaos.org/about/papers/advistmt/1026.asp
15. Computerized Physician Order Entry, http://en.wikipedia.org/wiki/Computerized_physician_order_entry
16. Radley, D.C., et al.: Reduction in medication errors in hospitals due to adoption of computerized provider order entry systems. J. Am. Med. Inform. Assoc. (February 2013)
17. PDRHealth, Physicians' Desk References, http://www.pdrhealth.com/drug_info/
18. Bar-coded medication administration, http://healthit.ahrq.gov/ahrq-funded-projects/emerging-lessons/bar-coded-medication-administration
19. Poon, E.G., et al.: Effect of Barcode technology on the safety of medication administration. New Engl. Journal Medicine (May 2010)
20. Montesi, G., Lechi, A.: Prevention of medication errors: detection and audit. J. Clin. Pharmacol. (June 2009)
21. Schneider, R., et al.: Bar-Code Medication Administration: A Systems Perspective. Am. J. Health Syst. Pharm. (2008)
22. Koppel, R., et al.: Workarounds to barcode medication administration systems: their occurrences, causes, and threats to patient safety. J. Am. Med. Inform. Association 15(4) (July-August 2008)
23. Kuperman, G.J., et al.: Medication related clinical decision support in computerized provider order entry systems: A Review. J. Am. Med. Inform. Assoc. (2007)
24. Koppel, B., et al.: Role of Computerized Physician Order Entry Systems in Facilitating Medication Errors. Journal of AMA 293(10) (2005)
25. Kawamoto, K., et al.: Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. BMJ (April 2005)
26. Rx Showcase, http://www.rxinsider.com/prescription_dispensing_automation.htm
27. Pyxis point-of-care verification, http://www.carefusion.com/medical-products/medication-management/point-of-care-verification/, and Pyxis Medication Station http://www.carefusion.com/medical-products/medication-management/medication-technologies/pyxis-medstation-system.aspx
28. Tsai, P.-H., Shih, C.-S., Liu, J.W.-S.: Mobile reminder for flexible and safe medication schedule for home users. In: Jacko, J.A. (ed.) Human-Computer Interaction, Part III, HCII 2011. LNCS, vol. 6763, pp. 107–116. Springer, Heidelberg (2011)

29. Tsai, P.H., Yu, C.Y., Shih, C.S., Liu, J.W.S.: Smart medication dispensers; design, architecture and implementation. IEEE Systems Journal (March 2011)
30. Tsai, P.H., Yu, C.Y., Wang, W.Y., Zao, J.K., Yeh, H.C., Shih, C.S., Liu, J.W.S.: iMAT: intelligent medication administration tools. In: Proceedings of IEEE International Conference on E-Health (July 2010)
31. e-pill, Pill Dispensers for Home and Institutional Use, http://www.epill.com/pillstation.html
32. Automatic Pill Dispensers with Alarms, http://www.dynamic-living.com/automated_medication_dispenser.htm
33. My Pill Box, http://www.mypillbox.org/mypillbox.php
34. Liu, J.W.S., Shih, C.S., Tan, C.T., Wu, V.J.S.: MeMDAS: medication management, dispensing and administration System. Presented at Mobile Health Workshop (July 2010)
35. Tsai, P.H., Chuang, Y.T., Chou, T.S., Shih, C.S., Liu, J.W.S.: iNuC: an intelligent mobile medication cart. In: Proceedings of the 2nd International Conference on Biomedical Engineering and Informatics (October 2009)
36. SISARL (Sensor Information Systems for Active Retirees and Assisted Living), http://sisarl.org
37. Tsai, P.H., Shih, C.S., Liu, J.W.S.: Algorithms for scheduling interactive medications. Foundations of Computing and Decision Sciences 34(4) (2009)
38. Tsai, P.H., Yeh, H.C., Yu, C.Y., Hsiu, P.C., Shih, C.S., Liu, J.W.S.: Compliance Enforcement of Temporal and Dosage Constraints. In: Proceedings of IEEE Real-Time Systems Symposium (December 2006)
39. Hsiu, P.C., Yeh, H.C., Tsai, P.H., Shih, C.S., Burkhardt, D.H., Kuo, T.W., Liu, J.W.S., Huang, T.Y.: A general model for medication scheduling. Institute of Information Science, Academia Sinica, Taiwan. Technical Report TR-IIS-05-008 (July 2005)
40. Yeh, H.C., Hsiu, P.C., Shih, C.S., Tsai, P.H., Liu, J.W.S.: APAMAT: A prescription algebra for medication authoring tool. In: Proceedings of IEEE International Conference on Systems (October 2006)
41. Wang, W.Y., Zao, J.K., Tsai, P.H., Liu, J.W.S.: Wedjat: A mobile phone based medication reminder and monitor. In: Proceedings of the 9th IEEE International Conference on Bioinformatics and Bioengineering (June 2009)
42. Liu, J.W.S.: Real-Time Systems. Pearson (2000)
43. Lin, H.C.: Intelligent monitoring, alert and notification tool. MS thesis, Department of Computer Science. National Tsing University, Taiwan (2011)
44. iNuC code, MeMDAS code, http://www.openfoundry.org/of/projects/1530
45. Pandey, S., et al.: Workflow engine for clouds. In: Cloud Computing, Principles and Paradigms. Wiley Series on Parallel and Distributed Computing (2011)
46. WfMC: Workflow Management Coalition, http://www.wfmc.org/
47. XPDL (XML Process Definition Language) Document (October 2005), http://www.wfmc.org/standards/docs/TC-1025_xpdl.2.2005-10-03.pdf
48. Pajunen, L., Chande, S.: Developing workflow engine for mobile devices. In: Proc. of IEEE International Enterprise Distributed Object Computing Conference (2007)
49. Bukovics, B.: Pro WF: Windows Workflow Foundation in.Net 4.0. Apress (2009)
50. Chen, T.-Y., et al.: Model-Based Development of User-Centric Automation and Assistive Devices/Systems. IEEE Systems Journal 6(3) (2012)
51. John, B.E., Kieras, D.E.: The GOMS family of user interface analysis techniques: comparison and contrast. ACM Trans. Computer-Human Interaction (1996)
52. Card, S.K., et al.: The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates (1983)