Information and Communication Technologies for Lifestyles of Health and Sustainability

Jane W. S. Liu[1] Chi-Sheng Shih[2]

[1]Institute of Information Science Academia Sinica, Taiwan [2]Dept. of Computer and Information Engineering National Taiwan University

【**Abstract**】This paper first presents examples of intelligent tools and consumer electronics designed specifically for the purpose of improving the quality of life and self-reliance of elderly individuals and people with chronic conditions. Some of them can also help all people in general to maintain life styles of wellness and enhance their sense of wellbeing. There are many challenging problems in making these devices and services, including how to make the devices safe, affordable and easy to use and maintain. The paper then discusses our approaches to solving these problems.

【**Keywords**】User-centric automation devices、tools for prevention of medication errors、smart device architecture、embedded workflow framework

1. Introduction

This paper describes recent efforts and accomplishments of faculty members and students from Academia Sinica, National Taiwan University, Tsing-Hua University and Chiao-Tung University who have been working together since the three-year (2006-2008) Academia Sinica thematic project SISARL (Sensor Information Systems for Active Retirees and Assisted Living). Information on the project and our current work can be found SISARL home page.

A thrust of our research is directed toward advancing the ICT (Information and Communication Technologies) for building low-cost, high-quality User-Centric Automation and Assistive Devices and Systems/Services (UCAADS). Examples of UCAADS include smart medication dispensers, autonomous home appliances and robotic helpers [1-11]. Some of the devices are specifically for the purpose of improving the quality of life and self-reliance of elderly individuals and people with chronic conditions or functionally limitations. Some of them can also help all people in general to live life styles of wellness and enhance their sense of wellbeing. Other UCAADS are automation and point-of-care tools (e.g., [12-15]) for enhancing the quality and reducing the costs of health and medical care by hospitals and other types of care-providing institutions.

We have been motivated by the same forces as many similar projects (e.g., [16-22]) that focus on technologies for elderly care and assisted living: We work to meet the needs for information and communication technologies that can offer effective solutions to the problems arisen from aging population, and we want to help our ICT industries exploit market and business opportunities offered by increasing demands of the elderly.

Our primary goals and underlying assumptions differ from many related efforts, however. Figure 1 illustrates the difference. Many projects assume that wellness and vitality of people beyond retirement age decline rapidly as illustrated by Part (a). Literatures on elderly-care technologies based on this assumption equate the elderly with people who are weak in the mind and body, in need of continuous monitoring, incapable of making simple decisions in daily life, and poorer than their young relatives and friends. The objectives of projects based on this view of the elderly are to develop monitoring and care-delivery tools and services for the purpose of managing their decline and reducing the costs of care delivery.

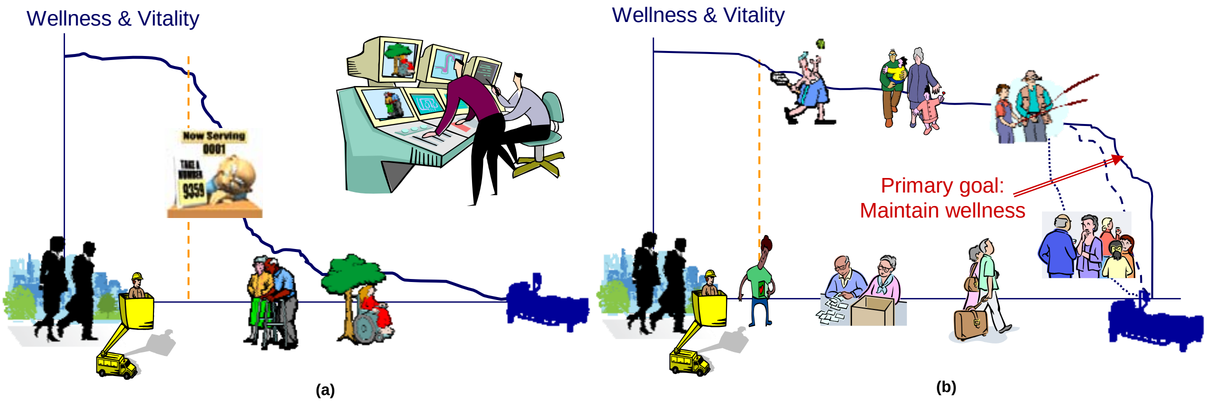

Figure 1 Project assumptions and goals

The view of the elderly illustrated by Part (a) leads to assumptions on usage scenarios and user requirements, and hence prototypes, that are inconsistent with the needs, desires, values and usage constraints of the majority of today's elderly. The fact is that improved health maintenance has not only lengthened life expectancy, but also has significantly increased the time span of the healthy and active part of old age. Vast majority of elderly individuals are of sound mind, can live and want to live independently, value their independence and privacy, and prefer to age in place. Without financial burdens of their youth, today's elderly individuals are typically as financially sound as ever in their life. It should not be a surprise that the total spending of this consumer group begins to exceed that of college-age consumers. Our projects are based on this assumption. Our goals are develop ICT that can help active elderly persons specifically, and busy people of all ages, to maintain wellness and vitality.

Despite vast differences in their functionalities and appearances, UCAADS share many common requirements. First and foremost, the devices must be *flexible*. A flexible device can be easily configured by its user(s) to work with different support infrastructures and operating conditions and can be customized to suite the user's preferences. Another common and critical requirement of UCAADS is that they must be safe: A *safe device* never does any harm and all unavoidable errors are either recoverable or tolerable. Ensuring that UCAADS are safe is challenging because many of these devices are semi-automatic: Rather than doing everything automatically for the user, the device may rely on the user to perform some mission-critical functions. The devices being user-centric and for discretionary use, their users typically have little or no training. Moreover, their skills vary widely among the user population and for an individual user over time.

Following this introduction, Section 2 describes a few UCAADS for illustrative purpose. Section 3 describes our approach to making flexibility a primary consideration in design, architecture and implementation of the devices. The section also describes our approach and current work on the problem of how to make devices safe. Section 4 summarizes the paper and presents our future work.

2. Representative UCAADS

We have designed and implemented several proof-of-concept UCAADS prototypes. This section describes some of them, including tools for prevention of medication errors and smart devices for convenience and safety.

(A) Intelligent Medication Administration Tools (iMAT)

Motivation and Overview Device and information technology for error-free medication administration has been a major focus of our research. This work has been motivated by alarming statistics on occurrences and consequences of preventable medication errors [23, 24]. Our medication dispenser, schedule manager and prescription authoring tool (collectively called iMAT) [7, 8] are designed to reduce administration errors (i.e., errors due to failure to stay compliant to correct medication directions). Such errors contribute a significant percentage (25–40%) of all preventable medication errors. A user targeted by iMAT may be on multiple interacting medications, some over long durations, and may have many prescriptions each year, in addition to over the counter (OTC) drugs and health supplements. Staying compliant over years and decades is challenging, even for healthy, active, and disciplined users.

Some of the most frequently cited reasons for non-compliance include inability to understand directions and inconvenience of rigid schedules. Existing administration and compliance aids (e.g., [25-27]) are of little or no help: One must manually load the medications into the dispenser, understand their directions and program the device accordingly. Modern drug libraries (e.g., [28]) provide comprehensive information on firm and hard dose size and timing constraints and compliance criteria for common medications. Because such information is not captured and 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.

iMAT dispenser is designed to eliminate these common causes of administration errors: The dispenser schedules individual doses of the user's medications under its care based a *medication schedule specification* (MSS). The machine readable specification is compiled from the user's prescriptions and directions of over the counter (OTC) drugs. The dispenser reminds the user at times when some doses should be taken, monitors user's response to reminders, adjusts the medication schedule as needed when the user is tardy, helps the user retrieve the right doses of right medications when user responded, and provides instructions on how the doses should be taken (e.g., with 8 oz of water, no food within 30 minutes, etc.). When the user is so tardy that non-compliance becomes unavoidable, the dispenser sends notification in ways specified by the MSS. In this way, iMAT helps the user follow medication directions and stay compliant without having to understand the directions.

Figure 2 shows how iMAT fits in a tool chain of information systems for medication use process: Specifically, iMAT sits at the bottom of the chain (in the lower half of the figure) supports the dispensing and administration stages of the process. It complements the tools for ordering and transcription stages, which include computerized physician order entry (CPOE) systems, clinical decision support (CDS), and electronic patient health and medication records (ePHR and eMAR) systems [28-33]. As the figure shows, iMAT interfaces with these tools through prescriptions and medication directions: iMAT takes the directions for the user as input and compiles them into a machine readable and formal specification for the medication scheduler. Part of the specification is illustrated by Figure 3.

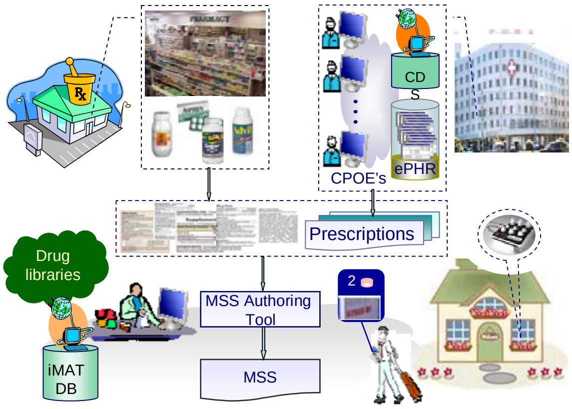

Figure 2 iMAT in medication use tool chain

Dosage Parameter Part of the MMS segment for medication M

1. *M*: Name of the medication
2. *g*: Granularity of dosage
3. [T_{min}, T_{max}]: Minimum and maximum durations
4. [d_{min}, d_{max}]: Nominal minimum and maximum dose sizes
5. [s_{min}, s_{max}]: Nominal minimum and maximum separations
6. (*B*, *R*): Maximum total intake over a specified time interval, given by budget *B* and replenishment delay *R*
7. (*L*, *P*): Minimum total intake over a specified time interval given by lower threshold *L* and a period length *P*
8. [D_{min}, D_{max}]: Absolute minimum and maximum dose sizes
9. [S_{min}, S_{max}]: Absolute minimum and maximum separations
10. *F*: Relationship with food
11. List of non-compliance event types and corresponding actions

Figure 3 A segment of MMS for a medication

Key assumptions iMAT makes two assumptions. The first is that all medications taken by each user are managed by a single dispenser (or a schedule manager). It is easy to see that no tool, not just iMAT, can be effective in preventing errors when this assumption is not valid.

The second assumption is that the user gets all of his/her medication supplies from some care provider who has full knowledge of all medications taken by him/her: To be concrete, we say here that it is the user's pharmacist. A user may have many prescriptions ordered by multiple physicians via independent CPOE systems and may take some OTC drugs at the same time. The pharmacist provides the last step in ensuring that the user's medications directions are correct. Whenever the user comes to get medication supplies, the pharmacist checks the user's medication directions with the help of a MSS authoring tool such as the one described in [2]. The tool helps the pharmacist detect drug interactions that may have left unaccounted for by some of user's medication directions. After resolving such problems and verifying that it is safe for the user to take the medications as directed, the tool generates a new MSS for the user's dispenser (or schedule manager). The pharmacist gives the user the MSS in a flash memory, along with new supplies, each medication in a separate container tagged with the RF id of the medication. To put new supplies of medications under the care of the dispenser, the user only needs to load the new MSS into the dispenser and plug the new containers into sockets on the dispenser base that holds the containers. We will return to provide more details.

Medication Schedule Specification A distinguishing feature of iMAT is the capability of its medication scheduler to take advantage of the scheduling flexibility provided by the user's medication directions: It can automatically adjust the schedule to help the user stay complaint whenever possible on occasions when the user is irresponsive to reminders. This is made possible by the general medication scheduling model [4, 34] underlying the MSS and the representation of medication directions formally in terms of scheduling constraints. Given the constraints, the iMAT medication scheduler can use algorithms described in [8, 35] to compute an initial schedule, and later adjust the schedule, meeting the constraints.

Figure 3 shows some of the constraints for a medication (say *M*) to help us explain. In general, for each medication *M* taken by the user, the specification contains a section consisting of parts extracted by the MSS authoring tool from an XML file on the medication in the iMAT database. The section 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 dispenser needs this information to manage and schedule the medication. The *dosage parameters* (DP) part in the section defines firm and hard 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; this part specifies changes in dosage parameters and additional timing constraints to account for the interactions.

Firm constraints are typically more stringent. The scheduler tries to meet all of these constraints whenever possible. Violations of firm constraints can occur nevertheless, usually due to tardiness or forgetfulness on the user's part. They degrade the quality of the schedule but may be acceptable. Firm constraints are defined by the sets of parameters in lines 4-7 in Figure 3: *Nominal dose size range* bounds the sizes, in term of multiples of granularity of *M,* of individual doses of M. *Nominal separation range* bounds the length of time between two consecutive doses of the medication. *Nominal maximum rate* (*B, R*) of M constrains the total size of all doses within any time interval of length *R* to be no more than *B*. Finally, *nominal minimum rate* (*L, P*) constrains the total size of all doses within any interval of length *P* to be at least equal to *L*.

Take Acetaminophen (Tylenol) as an example. Its direction 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. Since the drug is to be taken as needed, there is no required minimum total dose size for this drug; hence $(L, P) = (0, 24)$. In contrast, it is best not to skip any dose of a drug that should be taken one tablet every 8 hours for hypertension. Nevertheless, the direction may allow an occasional missed dose. This can be specified as $(L, P) = (2, 24)$ or (20, 168) (i.e., no more than one missed dose per day or per week).

Hard constraints of each medication *M* is specified in part by the two sets of parameters in lines 8 and 9 in Figure 3: The size of every dose must be in the *absolute dose size range* and the time separation between consecutive doses must be in the *absolute separation range*. Indeed, directions of almost all medications provide instructions in case "if you miss a dose". This instruction invariably leads to a wider absolute separation range $[S_{min}, S_{max}]$. 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.” Our experiments with scheduling real-life and synthetic sample prescriptions demonstrate that the more relaxed separation constraint can make schedules of most medications more tolerant to user tardiness and hence friendlier to the user [35].

We refer to a medication (or food) that interacts with *M* to the extent to require some changes in how *M* is to be administered as an interferer of *M.* The SI part of *M* has an entry for each of its interferers. The dosage parameters of M may need to be changed because 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 ranges. As an example, the constraint that an antibiotic should be taken on empty stomach (i.e., one hour before or two hours after meal) is specified by σ_{min} (Antibiotic, Food) = 1 and σ_{min} (Food, Antibiotic) = 2.

Components and Configurations Figure 4(a) and (c) show a picture of a stand-alone smart dispenser for home use and its internal structure, respectively. The hardware components are shown in the dotted box in the bottom half of Figure 4(c). As we can see, this dispenser has a memory card reader (i.e., a MSS port) for reading the MSS. The sockets on the base hold medication containers. Each socket is encircled by an indicator light. Other parts that interact with the user include a LED display, PTD (Push-To-Dispense) button, verification boxes, and dispensing cup, as well as an alarm device inside the base for delivering reminders locally. Also inside the base are an RFID reader and an array of switches (i.e., binary sensor array). There is a switch at the bottom of each socket for sensing whether the socket is empty or not.

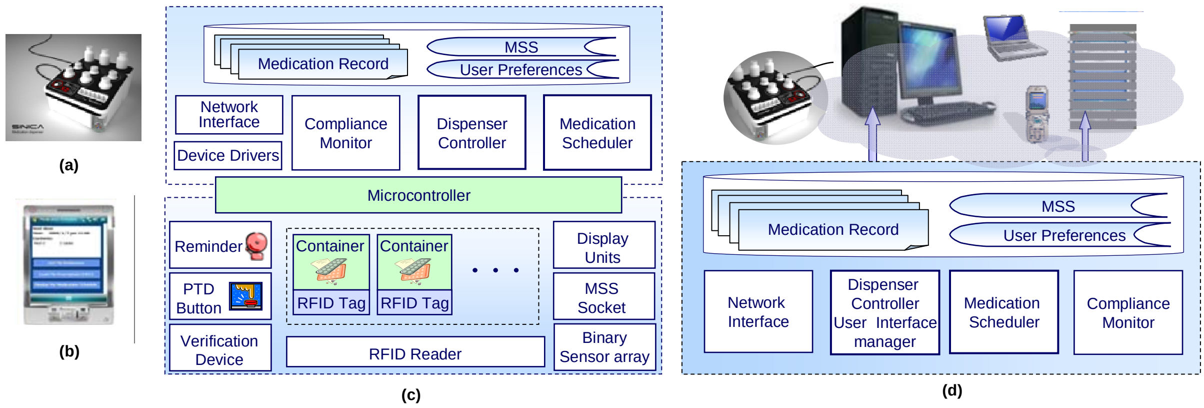

Figure 4 Dispenser and Scheduler Manager

The top half of Figure 4(c) shows the major software components. The interface to a dial up or broadband connection enables the dispenser to deliver reminders to the user remotely via a phone or a PDA and to send notifications of non-compliance events. The important work done by the device is divided between the (medication) scheduler and (dispenser) controller. The scheduler has full knowledge of what medication administration related actions should be done at what times. It does not keep track of time, however. That important task is done by the controller: The controller keeps track of time, informs the scheduler the arrival of the time instant for each action requested by the scheduler, and enables and monitors the execution of the action. Indeed, the controller monitors and controls the state of the dispenser. Details on how the operations of the dispenser and the structure of software system can be found in [7].

An advantage of the iMAT dispenser is that it can help, as much as an IT device can, to make sure that user retrieves the right dose of each medication from the right container when the user responds to reminder and comes to retrieve the medication. When the user is away from home and carries the medications with him/her on the road, the dispenser can provide reminders to the user by sending text and voice messages, with or without pictures of the medications, to his/her phone. This is what the picture in the lower right corner of Figure 1 tries to show. In case where the dispenser is connected to Internet, the user can acknowledge the receipt of each reminder and report his/her action taken as response to the reminder. With the user's permission, the compliance monitor can log all compliance related events. In this way, the dispenser can still carry out compliance monitoring function, though to a limited extent.

Some users do not need or want to have RFID tagged containers and the associated hardware. The mobile-phone based schedule manager and monitor provide them with most of the scheduling management and compliance monitoring functions of the iMAT dispenser. Part (b) of Figure 4 shows the portable schedule manager. The version described in [8] is a stand-alone device that does all the work of scheduling, delivering reminders, monitoring user response, etc. A user may choose to have his/her medication supplies delivered and use the pharmacy service of the supplier for downloading the MSS via Internet. A user who chooses to pick up supplies in person can have his/her MSS loaded to manager by the pharmacist.

Part (d) of Figure 4 shows the most flexible of available configurations. In essence, the schedule manager software runs on a PC or a laptop and uses one or more mobile devices for its interaction with the user. A user may start with only these parts. As he/she starts to spend more time at home and take more medications, the user can get a dispenser, less the software components, and connect it to the computer as a peripheral device. We have not yet implemented this configuration of iMAT.

Supporting Tools and Available Prototypes In addition to the end-user tools, iMAT also provides the MSS authoring tool for use by user's pharmacist. The tool helps the pharmacist process all of the user's medication directions to make sure that all drug interactions have been correctly accounted for before merging the directions and translate them into user's MSS. This tool requires the support of iMAT database. For sake of proof of concept, iMAT database now contains XML specifications of around 150 commonly used drugs. The specification of each drug provides the values of constraint parameters described earlier. For iMAT to be adopted and used in practice, we will need to expand the database to include at least all the medications available in Taiwan. To accomplish this initial goal and to maintain the database in the future, we need techniques and supporting tool(s) to automate the process of translating human readable directions in available drug libraries into XML specifications.

The source code of a dispenser controller prototype on Microsoft Windows XP and medication scheduling algorithms described in [35] have been released under BSD and GPL licenses on http://www.openfoundry.org/en/. They are listed as projects dispenser2 and medscheduler, respectively. We also have a Wedjat prototype. The next step is to redesign and architect the dispenser software so that a user can choose to have any one of the configurations shown in Figure 4 (d).

(B) Flexible System of Medication Management, Dispensing and Administration Tools

In addition to medication administration tools for naïve users, our efforts on ICT for medication error prevention include design and prototyping of error prevention tools for use by pharmacy and nursing staffs in care-providing institutions. This work is also motivated by the high rate of medication administration errors: The rate of error in name, frequency, dose size, sequence and time of medications given to patients can be as high as one in 5 to 6 doses, even in prestigious, world-class hospitals.

MeMDAS and Its Capabilities We are currently collaborating with nursing, pharmacy and IT staffs of National Taiwan University Hospital (NTUH) in the design and prototyping of a distributed system of smart medication cabinets and mobile nursing carts. The system is called MeMDAS (Medication Management, Dispensing and Administration System). The primary users of MeMDAS are pharmacy and nursing staffs in hospitals, long term care, and assisted living facilities.

Part (a) of Figure 5 shows how MeMDAS fit in the medication use process tool chain. It is designed to provide nursing and pharmacy staffs with the following tools and capabilities:

- Medication delivery and inventory capabilities for containing medication use costs;
- Bar-code controlled medication dispensing (BCMD) and bar-code controlled medication administration (BCMA) tools that have been shown to be effective in reducing rates of medication dispensing and administration errors [39];
- Modern work and time management (WTM), calendar and information access tools to help the nurses make the medication preparation and administration schedules of their patients central to their workday plans;
- Tools and user interface functions customizable to support automation to the degrees chosen by the users;
- 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 medication replenishment; and
- Customizable intelligent monitor, alert and notification (iMAN) tool to provide the capability of detecting event and action sequences that have a high likelihood to cause errors and alerting the user to take action to prevent them.

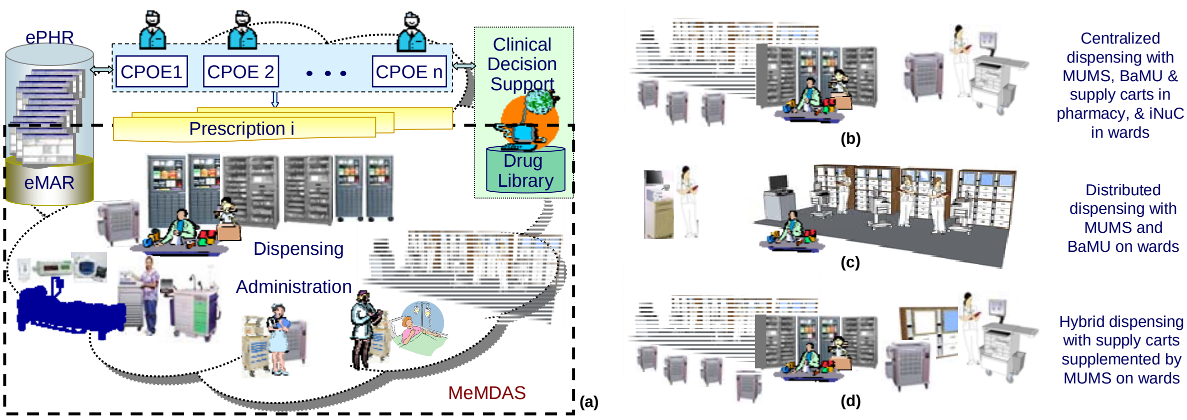

Figure 5 MeMDAS and Its Alternative Configurations

Multi-User Medication Station (***MUMS***) MUMS is a key component of MeMDAS. The term medication station refers to a system of smart cabinets controlled by a server. The capability of supporting concurrent dispensing to multiple users from the same MUMS is a novel feature. Medication stations by leading hospital automation equipment companies (e.g., [36-38]) typically operate in fully automated mode: When a user comes and selects 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 offset the advantages of using the station.

In contrast, MeMDAS medication stations are designed to be configurable so that they can operate in semi-automatic mode as well as fully automatic mode. When operating semi-automatically, the station server collaborates with the users and their mobile carts. Together, they enforce bar-code controlled dispensing in order to ensure the correct dispensing of medications to multiple users at the same time. Details on their interactions during this collaborative process can be found in [40].

Mobile Carts MeMDAS has two types of mobile carts: iNuC (intelligent Nursing Cart) and BaMU (Basic Mobile Unit). Unlike state-of-art mobile nursing carts, the software system controlling their operations can be easily configured to make an iNuC works as a BaMU and vice versus. Except for BCMD, the beta version of iNuC, called iNuC 1.0, offers its user (a nurse) all the capabilities listed above without help from MUMS, and, in events of network and hospital information system outages, can function stand-alone. The newer versions of iNuC currently being implemented can also collaborate with MUMS servers to enforce BCMD.

iNuC 1.0, is ready for trial use. It is written in C and C# and runs on Microsoft

XP Embedded. Its source code is released under GPL license and can be downloaded from http://of.openfoundry.org/projects/1140.

A BaMU is a light-weight system of mobile tools for use by nurses during bar-code controlled dispensing of medications from MUMS. 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 provided by iNuC. Some of such BaMU's are used in wards that have computers at patients' bedside for these purposes. Such a BaMU can also function as an intelligent medication supply cart for use by pharmacy staff.

Alternative Configurations A distinguishing characteristic of MeMDAS is configurability and customizability. In particular, components of MeMDAS can be easily configured and used to support centralized, distributed and hybrid dispensing. In a hospital using centralized dispensing process, the pharmacy prepares and delivers daily to each ward a supply cart with drawers. The daily doses of medications for each patient in the ward are in one or more drawers. As part (b) of Figure 5 shows, one or more MUMS can be used in the pharmacy together with BaMU's as intelligent supply carts. Together, they support BCMD and make the manual part of that supply-cart preparation process less error prone.

Distributed dispensing process is often used for departments and wards (e.g., ICU and OR) where patients' prescriptions change frequently. Part (c) of Figure 5 depicts the use of MUMS and BaMU in such wards to provide control and safeguard: The pharmacy monitors and stocks the cabinets in the station with all or most medications needed for patients in the 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 the nurse's BaMU. Carts may not be allowed in patients' rooms. The figure shows this case: The nurse brings the locked drawer of each patient to the patient's bedside after filling the drawer with the doses due to be administrated and uses the BCMA tools provided by the bedside computer to assist the administration of the medications.

Distributed dispensing tends to increase workload for nursing staff. This is why hybrid processes are common. As Part (d) of Figure 5 illustrates, some medications are dispensed and delivered via supply carts by the pharmacy department. Some patient wards also have medication stations and use them to hold controlled drugs and frequently used medications, making it possible for nursing staffs to get newly ordered medications on a timely basis.

In wards where dispensing is centralized or hybrid, nurses uses iNuC for BCMA: To put the medication drawer of a patient under the control of an iNuC, the nurse removes the drawer from the supply cart, scans the bar-code 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. In this way, the cart acquires the association between the id of the new drawer, its location in the cart and patient's bar-code id. From this information, it creates the mapping between the drawer location and the patient id. When the patient is due to take some medication(s), the nurse can open the patient's drawer at bedside by scanning the patient's bar-code id in the wristband worn by the patient.

(C) Smart Devices for Quality of Life and Safety

The UCAADS prototypes described below were developed within the SISARL project. They were designed for active retirees, but can benefit active and busy people of all ages.

Smart Storage Pantry By a storage pantry, we mean specifically a shelf or open cabinet used to hold non-perishable, household supplies, such as shampoo, detergent, cans and extra supply of beer. Parts (a) and (b) of Figure 6 show how such a pantry may look when objects of each kind are placed together in an actual or virtual compartment. As the figure illustrates, many objects are heavy, bulky, and boring to shop for, yet are essential. For people who live in cities where apartments are small and discount stores are miles away, the chore of keeping essential supplies like these on hand can be annoying, even burdensome.

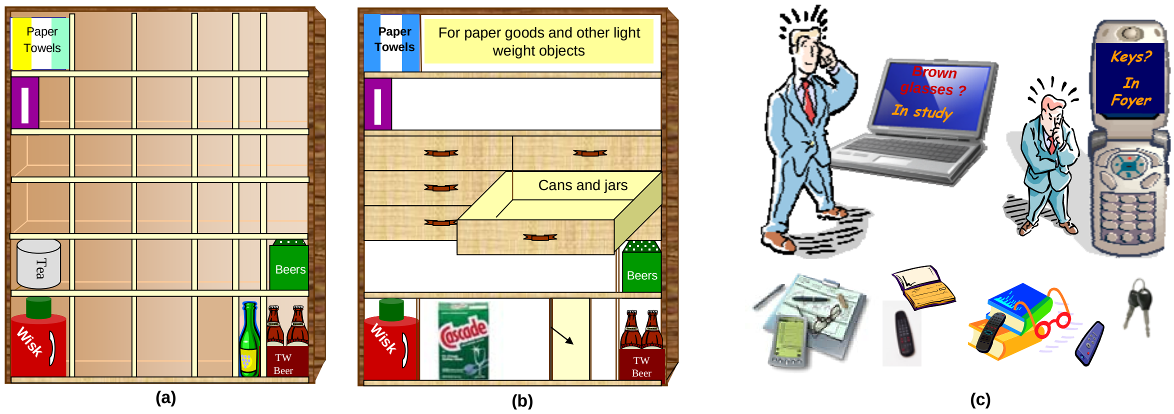

Figure 6 Sample devices of convenience: smart pantry and object locator

A smart pantry can determine from sensor readings when the last unit of any supply is removed, leaving the compartment occupied by the object empty. Sensing this condition, the pantry automatically contacts the store designated previously by the user, places an order for a specified quantity of the supply, and arranges payment and delivery of the supply within a specified length of time. In this way, the pantry and supporting delivery service relieve the user from the need to shop or order such supplies. An additional advantage is energy saving from reduced travel of pantry owners to and from discount stores.

Clearly, a smart pantry must have some means to identify and monitor its content. A pantry using RFID for this purpose is fully automatic and easy to use, but not economically feasible: Even a cent per tag is too costly to replace bar codes on bottles of shampoo, rolls of papers, etc. We have built and experimented with a pantry that uses a digital camera for content capture [3]. It is also fully automatic and easy to use for the pantry owner, but its usability is poor for suppliers. Each picture in purchase orders from the pantry must be processed to identify the brand and size of the object in it. Because picture quality is not ideal and object search space is large, most object identification methods cannot attend the required error rate.

We also built a BAC (bar code) pantry. A BAC pantry identifies supplies in it by their bar codes. The prototype is also described in [3]. It has the best cost versus usability tradeoff today. In particular, it is ideal for suppliers since every bar code in orders from the pantry uniquely identify the item to be delivered. The pantry cannot work without user's help, however. The work required of the user (i.e., the owner) is actually simple: before placing an object into an empty compartment, scan its bar code. The scan-place activity of the user triggers the pantry to generate and maintain a compartment-id-bar-code association. When a compartment becomes empty, the pantry inserts the associated bar code in the purchase order. The process of acquiring bar codes is error prone. A busy user may dump supplies in the pantry without scanning their bar codes. Multiple users may put away objects and remove objects at the same time. We cannot restrict user-pantry interaction patterns but must make sure that the device works satisfactorily regardless. As mentioned earlier, this is a challenge common to most user-centric devices. The SISARL component model and simulation environment [41, 42] were motivated by the need to address this problem.

Object Locator An object locator is a device designed to assist its users in finding misplaced household and personal objects in a house or at work. A locator such as the ones offered by specialty stores contains an interrogator with several buttons of different colors and a tag of the color matching the color of each button. By attaching a tag to an object, the user can look for the object by pressing the button of matching color on the interrogator. The tag attached to the object beeps in response and thus enables the user to find the object.

The existing object locators are not ideal in many aspects: The number of buttons on the interrogator is fixed; extending the locator to track more than that number (e.g., 8-16) of objects is impossible. Tags are battery-powered. A tag might become a lost object itself after it runs out of battery. When a tag breaks, a user must purchase a replacement tag of the same color as the broken one. If a user were to use two tags of the same color, both tags would respond to the search signal from the interrogator. This situation is clearly not desirable.

We have implemented a proof-of-concept prototype of object locator based on the RFID (Radio Frequency Identification) technology [5]. The prototype can use a smart phone or a laptop for user interface and provides its users with the capability of querying for locations of misplaced objects (e.g., keys, reading glasses and important documents) by names. Rather than having the tag beep and flash, the locator responds by displaying the location of the queried object. Part (c) of Figure 6 illustrates this concept. In addition to the prototype, we evaluated alternative designs of device. The differences in their hardware capabilities and object search schemes lead to differences in search time and energy consumption.

RFID-based object locators based on all the designs are extensible, reusable, and low maintenance. They are extensible in the sense that the maximum number of tracked objects is practically unlimited and that a RFID-based object locator can support multiple interrogators. The interrogator software can run on a variety of platforms (e.g. desktop PC, PDA, smart phone and so on). Reusability results from the fact that all RFID tags used for object locators can have globally unique ids. Hence, tags never conflict, and a tag can be used in more than one object locators. Low maintenance is one of the advantages of RFID technology. The version shown in Figure 6 reports the room where the misplaced object is found. When the user goes to room, he/she must still search for the object in the room or if the tag attached to the object can beep, commands it to beep. The proof-of-concept prototype displays object location with accuracy of 1-2 meter. It uses only passive RFID tags. A user of a locator based on this designed is never burdened by the concern that a tag may be out of battery.

The potential world-wide market of this device is huge: billions of readers and tags. The RFID readers and tags ideal for object locator application differ from current readers and tags, which are optimized for applications such as inventory control and commodity flow. For example, object locator calls for medium and small range readers and tags with omni-directional antennae. Cost per reader needs to be two orders of magnitude lower than current price and the query signal from readers needs to charge batteries of passive tags sufficiently to enable the tags to beep. On the other hand, some requirements of object locator are much more relaxed compared with most RFID applications: Readers always query tags in address-mode and can take a second or two to read a tag. RFID industry should be able to take advantage of the relaxed requirement to reduce reader cost and boast tag battery energy.

Walker's Buddy A walker's buddy is a device designed to be worn by a walker or jogger. It can detect and warn the wearer of uneven pavement ahead and thus help preventing falls. Part (a) of Figure 7 illustrates how such a device may be used. In this picture, the buddy is a part of a smart phone or some other hand held device. The user carries it at the waist (e.g., in a phone case.) As the user walks, the device irradiates and illuminates the path ahead. It warns the user when it detects steps, bumps, or other hazards in the path. Here, the user is a busy office worker walking to work or returning home. One also envisions an active elderly individual carrying it during morning or evening walks on city sidewalks or parks.

Figure 7 Walker's buddy and sensor devices for sidewalk safety

For the device to be useful, it must work during day light hours, dust and night.

It also must work anywhere. This requirement implies that the device must be self contained. In particular, it should not rely on any pedestrian and vehicular service (such as electronic bulletins and messages from some intelligent transportation and pedestrian safety system) along the path to provide it with information on where hazards are.

Figure 7(a) assumes that the device is capable of distinguishing steps up or down and tells the user the downward step ahead. In general, a device may not be able to distinguish the types of hazard and issue specific directives. It can only provide general warnings, such as “watch your step”, which a companion is likely to say when the user walks with friends. Indeed, the user should be able to choose among many forms of warning sound, including recorded voices of friends who often walk, stroll or jog together.

The prototype described in [43] makes use of an ultrasound range finder to detect obstacles or uneven pavement a few paces in front of the wearer. Its resolution is less than ideal: We are investigating alternative techniques that may yield better resolution and faster response.

In recent years, there are more and more bikers sharing sidewalks with joggers and walkers. Part (b) of Figure 7 illustrates scenes that are increasingly more commonplace. Sensor devices that can warn pedestrians of on-coming bikes and bikers to be aware of people in the road are useful for all people who walk and jog regularly. This is an area of our research interest.

3. Technologies for Building Flexible and Safe UCAADS

As mentioned earlier, challenges in building high-quality, low-cost UCAADS include how to model, architecture and build flexible UCAADS and how to ensure such a device never causes any harm and works acceptably well as it adapts to user’s condition and needs. Our research focus has been on these problems. A major thrust has been directed towards system architecture, components, platform and tool that support the workflow approach [44] to building flexible UCAADS. Our work on safe UCAADS focuses on an aspect that is unique for semi-automatic devices: One way to keep the system safe is to instrument the device so it can effectively monitor user actions and prevent the user(s) from causing unsafe operations and intolerable faults. For this, we need techniques and tools.

(A) Workflow Approach to Flexibility

We have adopted the workflow paradigm [44] as a way to make flexibility a design objective, from model, architecture and design of UCAADS and the framework for their development and evaluation. This paradigm has been widely used in enterprise systems for automation of business processes. Using this paradigm, the developer decomposes work to be done by the application into basic building blocks called *activities*. An activity may be done by executing a software procedure or a hardware driver controlling some device (e.g., a motor controller). We call such an activity a *software activity*. Other activities may be done by an operation of a hardware device, delivery of a message by a network link, and so on. In a semi-automatic device, some activities are actions taken by human users. We call these activities *external activities*. Activities are composed into module-level components called *workflows*. The order and conditions under which activities in a workflow are executed, the resources needed for their execution, and interactions among activities are defined by the developer of the workflow. The definition can be in terms of some programming language (e.g., C# in [45]), a process definition language (e.g., XPDL, WfMC standard XML Process Definition Language [46, 47]), or an execution language (e.g., BPEL, Business Process Execution Language [48]). Workflows can also be defined graphically (e.g. [45]): In a workflow graph, nodes represent activities in workflows or states of the device, and directed edges represent transitions between activities or between states.

Workflow-Based System Architecture Modern engines can handle not only automated process but also activities triggered by external events and can treat manual activities by users and automated activities by hardware and software in an integrated way. Using such an engine, we can build embedded device families based on a workflow-based architecture: Only a small part of a device with this architecture is hardwired. Most of it is built from activities and workflows components. The workflow engine integrates the components by executing them and arbitrating their resource contentions as specified by the workflow graphs.

A workflow-based application can be easily configured and customized by changing the workflow graphs in it and by invoking different components for activities in the graphs. Figure 8 illustrates this point. It shows the workflow-based structures of iNuC 1.5 and MUMS server, two major components of the medication management, dispensing and administration system described in the previous section: Both iNuC 1.5 and MUMS are being implemented to run on Microsoft Windows .NET 3.5 Workflow Foundations [45].

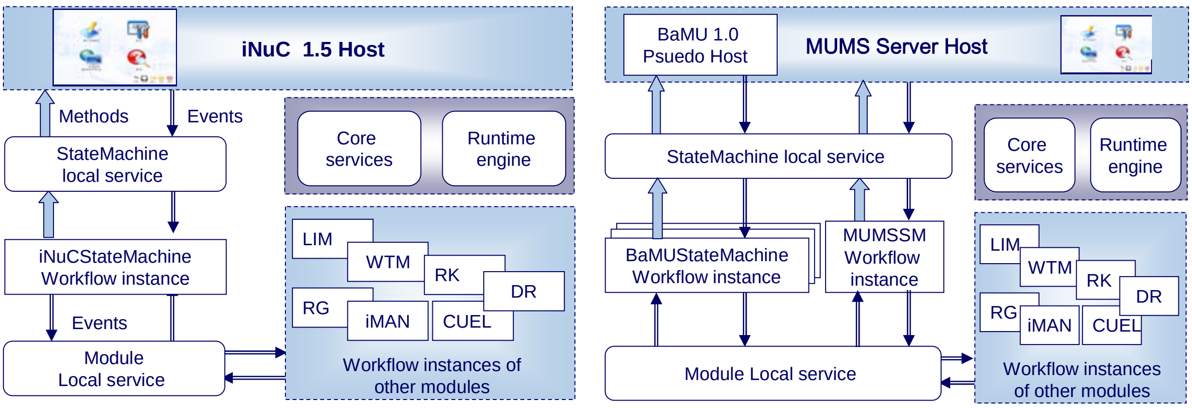

Figure 8 Workflow-based structures of MeMDAS component systems

Details in the diagrams are unimportant for our discussion here. It suffices to note here the similarity between the structures. Software components that handle work and time management (WTM), locker interlock mechanism (LIM), data refresher (DR), recorder keeper (RK), intelligent monitor, alert and notification (iMAN), and so on are compiled into dynamic link library (DLL) functions that are executed as activities in workflows of the modules.

As mentioned earlier, we want to be able to easily configure an iNuC into a BaMU. This is done by replacing the iNuC state machine workflow instance (in the left column in the diagram on the left) by a BaMU state machine workflow instance and iNuC 1.5 host thread by BaMU host. By reversing the change, we can configure the system from a BaMU to an iNuC. The MUMS server can support a few remote sessions via BaMU concurrently. This is why it may have multiple BaMU state machine workflow instances interacting via a local service (i.e., communication facility) with BaMU pseudo host and through it with GUI running remotely on basic mobile carts.

***EMWF* (*Embedded Workflow Framework*)** To date, few embedded devices and systems are built on workflow-based architecture. A major reason is the lack of engines suited for embedded systems, including many UCAADS such as iMAT dispenser. Memory space and processor bandwidth of these devices are typically not scarce. Still even they cannot accommodate the resource demands of existing engines designed to run in J2EE or .Net environments. Another reason is that the standard language XPDL (XML Process Definition Language) [47] is too rich than needed for definitions of embedded workflows on one hand. On the other hand, it lacks many important elements. Examples include means for specifying timing constraints and basic activities (e.g., behavior coordination) for some applications (e.g., behavior-based robots). For flexible applications, the restriction to static workflow graphs forces us to provide a priori workflows for different configurations and adaptive behaviors. The ability to modify workflow graphs dynamically at runtime is also a needed feature named by the scientific computing community.

We designed and prototyped EMWF (Embedded Workflow Framework) [49] to address these issues. EMWF is an open source middleware for flexible UCAADS. EMWF 1.0 provides a light-weight workflow manager and engines on Windows CE, Windows XP Embedded, and Linux. The process definition language supported by EMWF is called SISARL-XPDL. It consists of a subset of the WfMC standard XML Process Definition Language (XPDL) 2.0, together with elements that implement common mechanisms for robot behavior coordination. EMWF engines are written in C in order to keep their memory footprint and runtime overhead small. Performance data show that the overheads introduced by the engine and workflow data are tolerable.

Being a proof-of-concept prototype, this version has many limitations. For example, it lacks support for flexible and efficient communication between workflows. (This is the reason we are implementing MeMDAS to run on Windows WF.) We are working to remove these limitations. The future version EMWF 2.0 will provide basic message passing mechanism, real-time scheduling capability and workflow communication facility.

Executable Operational Specification In addition to using workflows as components of UCAAD, we use workflows for two other purposes. The first is using workflows for integration of embedded devices together and with support infrastructures. An example is the framework for integrating smart medication dispensers at the end-user level of the tool chain for medication use process with upper level tools [50].

The second usage of workflows is for *operational specifications* of device behavior, user actions and user-device interactions. According to the underlying SISARL component model [41], a device has a resource (structural) view specification and an operational (behavior) view specification. The formal tells us the components used to build the device and interconnections of the components. The operational (behavior) view specification tells us how the device works and what actions it expects from its user(s).

Being written in terms workflows, the operational specification of a device is executable. The developer of a new or modified device can use the operational specification of the device as input and have it executed in the SISARL simulation environment [42] in simulation experiments for the purposes of evaluating the device through out the development process.

(B) Monitoring Techniques and Tools

As stated earlier, we are concerned with how to keep the device and its user(s) safe as parts of the symbiotic system as a whole changes. There have been extensive works on user models, formal verification methods, and runtime monitoring techniques [51], and so on for this purpose. A typical assumption is that the device is used as intended. This assumption is usually valid for machine-centric devices with their well trained users, but is not valid for UCAADS without help for their users. We are exploring the idea of having the device monitor user actions at runtime and help to prevent misuses that may have serious consequences.

Take the iMAT dispenser as an example. An assumption for the set up operation is that medication containers are plugged into empty sockets one at the time. The dispenser can correctly locate the container and its content only when this assumption is true. In this case, the device can determine whether the assumption is valid by monitoring its own states. When it detects user actions that lead to violation of the assumption (e.g., seeing more than one socket changes state from empty to non-empty), it works with the user to correct the situation.

In a general runtime monitoring environment for this purpose, the device may have control mechanisms that do not interact with the user. The symbiotic system is modeled as a feedback loop which takes user actions as input. User actions are partitioned into three types: Assumed, Disallowed and Tolerable. The device guarantees to work as intended if the user actions are of the Assumed type. A device well designed for usability should have only few actions of the Disallowed type. A Disallowed action may cause serious malfunctions. In the case of iMAT dispenser, actions involving plugging containers are either Assumed or Disallowed. An easy to use device may allow many Tolerable actions. The device works in a degraded manner in this case. An example is a semi-automatic smart pantry. When the users act as Assumed, it can order the right supplies to be delivered just in time. A Tolerable action can cause the pantry to fail in a placing an order, but it is aware of the failure and can warn the user in time. A Disallowed action causes wrong supplies to be delivered, incurring expenses and annoyance.

Needless to say that such a monitoring scheme works only if device state changes caused by user actions are observable. To be effective, we also want the device to be able to rollback when a Disallowed action is observed, and in this sense, to be controllable. This is indeed the case for iMAT dispenser.   In the case of smart pantry, the device is not always able to distinguish types of user actions. For such devices, runtime monitoring does not work.

Thus far, we have been looking at this problem in an ad hoc manner, one device at a time, as in the case of MeMDAS iMAN component. General design principles such as disallowed actions should always lead to observable state changes cannot be easily translated into general design guidelines and rules and working methods and tools. Without them, we are forced to play it safe. This typically means that we restrict the adaptability and sacrifice usability of the device so we are sure it is always safe and often sound.

4. Summary

In the near future, we will continue to pursue technology transition of relative mature results, applied research on middleware and tools, and basic research on distributed, real-time workflow framework and run-time monitoring.

iMAT and iNuC 1.0 are both ready for experimental use and evaluation. A natural next step of iMAT is to conduct a field trial to determine its effectiveness and tune its usability. We plan to start this work in the coming year. iNuC 1.0 is being used experimentally by nurses in NTUH physical therapy department. In addition to exposing bugs that our own tests fail to find, they have pointed out many places where usability of the tools can be improved. Their feedbacks are being incorporated into iNuC 1.5 and other components of MeMDAS. The target completion date of MeMDAS prototype is January 2011. The next step is to have the system used on experimental basis by intensive care unit as well as physical therapy department.

As stated earlier, we are exploiting the workflow approach to design, building and evaluation of flexible UCAADS and use the paradigm in two novel ways: First, we use workflows as executable behavior specifications of devices and device-user interactions that can be used as inputs to simulation experiments. Second, we adopt workflow-based architecture and implementation for embedded devices and service robots.

In addition to EMWF described above, we are incrementally building a workflow-based simulation environment, now named USE (UCAADS Simulation Environment). USE provides the developers with several reusable component libraries, including a repository of specifications, code and executables which a developer of a new or modified device can use to put together from early design stage an executable model of the device. In addition, it has a library of user models to support experimentation and evaluation of the devices in real-life like use scenarios. Thus far, USE runs on Microsoft Windows .NET Workflow Foundation. The next step is to port the models and tools so that they also run on EMWF. In this way, we will turn USE from a pure simulation tool into a development environment that will enable the developer of a UCAADS to start from operational specification of the device, use the specification for evaluation purposes by executing it in simulation experiments within USE on WF or EMWF, and then have the specification run on the target platform and EMWF to produce an implementation of the device.

We have begun recently to investigate alternatives in algorithms and mechanisms for end-to-end resource and quality of service management in open environments that can make EMWF capable of supporting real-time systems abstractions needed by time critical UCAADS. This work intends to pave a solid foundation for a distributed, real-time workflow framework for time-critical cyber-physical systems in general.

Acknowledgement

Our efforts have been partially supported by Taiwan Academia Sinica thematic project SISARL, funds provided by Institute of Information Science and Research Center of Information Technology Innovation, and by Projects iNuC and MeMDAS supported by National Taiwan University Hospital.

Selected References

[1] J. W. S. Liu, B. Y. Wang, C. S. Shih, T. W. Kuo, A. C. Pang, T. S. Chou, W. H. Chen, Y. T. Liu, H. C. Yeh, , J. K. Zao, C. W. Hsueh, and C. Y. Huang. "User Scenarios and Designs of Smart Pantry, Object Locater, and Walker's Buddy," Institute of Information Science, Academia Sinica, Taiwan, Technical Report TR-IIS-05-007, July 2005.

[2] 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*, Vol. 5, pp. 4,284-4,291, October 2006.

[3] C. F. Hsu, H. Y. M. Liao, P. C. Hsiu, Y. S. Lin, C. S. Shih, T. W. Kuo, and J. W. S. Liu, "Smart Pantries of Homes," *Proceedings of IEEE International Conference on Systems*, *Man and Cybernetics*, Vol. 5, pp. 4,276-4,283,October 2006.

[4] P. H. Tsai, H. C. Yeh, C. Y. Yu, P. C. Hsiu, C. S. Shih and J. W. S. Liu, "Compliance Enforcement of Temporal and Dosage Constraints," *Proceedings of the 27th IEEE Real-Time Systems Symposium*, December 2006.

[5] T. S. Chou and J. W. S. Liu, "Design and Implementation of RFID-Based Object Locator," *Proceedings of IEEE RFID Technologies*, TuB2.2, pp. 86–93, March 2007.

[6] H. C. Yeh, C. S. Shih and J. W. S. Liu, "Integration Framework for Medication Use Process," *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, October 2007.

[7] P. H. Tsai, C. Y. Yu, W. Y. Wang, J. K. Zao, H. C. Yeh, C. S. Shih, and J. W. S. Liu, "iMAT: Intelligent Medication Administration Tools," *Proceedings of IEEE Healthcom*, July 2010.

[8] W. Y. Wang, J. K. Zao, P. H. Tsai, and J. W. S. Liu, "A Mobile Phone Based Medication Reminder and Monitor," *Proceedings of IEEE HealthCom*, July 2010.

[9] Forizzi, J. and C. DiSalvo, "Service Robots in Domestic Environment: a Study of Roomba Vacuum in the Home," Proc. of ACM/IEEE International Conference on HRI, March 2006.

[10] Kaneshige,Y., M. Nihei, and M. G. Fujie, "Development of New Mobility Assistive Robot for Elderly People with Body Functional Control," *Proceedings of IEEE/RAS-EMBS,* February 2006.

[11] Lin, C. H., Y. Q. Wang and K. T. Song, "Personal Assistant Robot," *Proceedings of IEEE International Conference on Mechatronics*, July 2005.

[12] SpeciMinder hospital delivery robot, http://www.youtube.com/watch?v=lJ7RnTAYZ-8

[13] TUG, pharmacy delivery robot, http://hfrp.umm.edu/tug/tug_main.htm

[14] Tsai, P. H., Y. T. Chuang, T. S. Chou, C. S. Shih, and J. W. S. Liu, "iNuC: An

Intelligent Mobile Medication Cart," Proceedings of the 2nd International Conference on Biomedical Engineering and Informatics, October 2009.

[15] Liu, J. W. S., C. S. Shih, C. T. Tan and V. J. S. Wu, "MeMDAS: Medication Management, Dispensing and Administration System," Mobile Health Workshop, July 2010.

[16] Dishman, "Inventing Wellness Systems for Aging in Place," IEEE *Computer,* May 2004 and Age-in-place, http://www.intel.com/research/prohealth/, Intel Corporation.

[17] Changing Places Consortium, http://architecture.mit.edu/house_n/, MIT.

[18] Aware home, http://awarehome.imtc.gatech.edu/, Georgia Tech.

[19] Center for Future Health, http://www.futurehealth.rochester.edu/, University of Rochester.

[20] Marc smart home, http://marc.med.virginia.edu/projects_smarthomemonitor.html, University of Virginia.

[21] Assisted Cognition Projects, http://www.cs.washington.edu/assistcog/, University of Washington.

[22] Bock, T., "Service Oriented Architecture-Robot Assisted Living," http://www.globalaging.org/elderrights/world/2008/service.pdf.

[23] "Drug Safety and Availability: Medication Errors." June 18, 2009. http://www.fda.gov/Drugs/DrugSafety/MedicationErrors/default.htm.

[24] E. G. Poon, et al. "Effect of Bar-Code Technology on the Safety of Medication Administration," The New England Journal of Medicine, Many 2010.

[25] e-pill, Pill Dispenser: http://www.epill.com/md2.html.

[26] Pill boxes: http://www.dynamic-living.com/automated_medication_dispenser.htm.

[27] My Pill Box at http://www.mypillbox.org/mypillbox.php.

[28] PDRHealth, Drug Information, http://www.pdrhealth.com/drug_info/.

[29] CPOE.org, http://www.cpoe.org/.

[30] Kuperman, G. J., A. Bobb, T. H. Payne et al. "Medication related clinical decision support in computerized provider order entry systems: A Review" *J. Am. Med. Inform. Assoc.,* 2007.

[31] 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.

[32] Electronic Medication Administration Record (eMAR), http://www.dh.org/body.cfm?id=434&oTopID=434.

[33] Health Information Systems: http://www.hhs.gov/healthit/ahic.html.

[34] Hsiu, P. C., 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.

[35] P. H. Tsai, C. S. Shih, and J. W. S. Liu, "Algorithms for Scheduling Interactive Medications," *Foundations of Computing and Decision Sciences*, Vol. 34, No. 4, 2009.

[36] Rx Showcase, http://www.rxinsider.com/prescription_dispensing_automation.htm.

[37] Pyxis Medstation, http://cardinal.com/us/en/providers/products/pyxis/brochure/MS3500_spec_details_PLA.pdf.

[38] InfoLogix Ultra Rx Medication Cart, http://healthcare.infologixsys.com/pictures/products/medication-cart-ultra-rx.pdf.

[39] E. G. Poon, *et.al*, "Effect of Bar-Code Technology on the Safety of Medication Administration," *The New England Journal of Med*, May 2010.

[40] J. W. S. Liu, C. S. Shih, C. T. Tan and V. J. S. Wu, "Medication Management, Dispensing, and Administration System," presented by U-Health Workshop, IEEE HealthCom, July 2010, available for download at http://www.sisarl.org/index.php?option=com_docman&task=cat_view&gid=22&Itemid=33.

[41] T.Y. Chen, P. H. Tsai, T. S. Chou, C. S. Shih, T. W. Kuo, and J. W. S. Liu, "Component Model and Architecture of Smart Devices for the Elderly," *Proceedings of the 7th Working IEEE/IFIP Conference on Software Architecture*, pp. 51–60, February 2008.

[42] T. Y. Chen, C. H. Chen, C. S. Shih, J. W. S. Liu, "A Simulation Environment for the Development of Smart Devices for the Elderly," *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, October 2008.

[43] Y. S. Hsu, S. F. Hsiao, C. E. Chiang, Y. H. Chien, H.-W. Tseng, A.C. Pang, T. W. Kuo, and K. H. Chiang, "Walker's Buddy: An ultrasonic Dangerous Terrain Detection Systems," *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, September 2006.

[44] Workflow definition, http://en.wikipedia.org/wiki/Workflow.

[45] Microsoft Windows .NET, Workflow Foundation, http://msdn2.microsoft.com/en-us/netframework/aa663328.aspx.

[46] WfMC: Workflow Management Coalition, http://www.wfmc.org/.

[47] XPDL (XML Process Definition Language) Document, http://www.wfmc.org/standards/docs/TC-1025_xpdl.2.2005-10-03.pdf , October 2005.

[48] BPEL (Business Process Execution Language),

http://en.wikipedia.org/wiki/BPEL.

[49] Chou, T. S., S. Y. Chang, Y. F. Lu, Y. C. Wang, M. K. Ouyang, C. S. Shih, T. W. Kuo, J. S. Hu and J. W. S. Liu, "EMWF for Flexible Automation and Assistive Devices," *Proceedings of IEEE Real-Time and Embedded Applications and Systems Symposium,* April 2009.

[50] H. C. Yeh, C. S. Shih and J. W. S. Liu, "Integration framework for medication use process," *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, October 2007.

[51] M. Viswanathan and M. Kim, "Foundations of the run-time monitoring of reactive systems," Theoretical Aspect of Computing, Vol. 3407/2005, Springer.

【作者簡介】

張韻詩　女

職　　稱：中研院 資訊科學研究所 講座研究員

研究領域：即時系統、嵌入式系統、自動化及輔助設備

個人簡介：麻省理工學院獲博士學位，曾任教於美國伊利諾斯大學計算機科學系達 27 年，出版過 2 本大學教科書，以及發表超過 170 篇的學術論文。研究專長爲即時系統、嵌入式系統、分佈式系統。近年來的研究工作也包括爲銀髮族設計的智慧型電子設備及服務等研究及先進的開發。

聯絡電話：886-2-27883799 ext.1807

聯絡郵箱：janeliu@iis.sinica.edu.tw