NRC·CNRC

NRC Publications Archive
Archives des publications du CNRC

A system for the seamless integration of personal messages using agents developed on a Lotus Notes platform

Liscano, Ramiro; Impey, Roger; Gordon, P.; Abu-Hakima, A.

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. / La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.
For the publisher's version, please access the DOI link below./ Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

https://doi.org/10.4224/9167749

NRC Publications Record / Notice d'Archives des publications de CNRC:
https://nrc-publications.canada.ca/eng/view/object/?id=cdc5efb7-ee78-46ce-b2b6-4099cc5a5506
https://publications-cnrc.canada.ca/fra/voir/objet/?id=cdc5efb7-ee78-46ce-b2b6-4099cc5a5506

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
https://nrc-publications.canada.ca/eng/copyright
READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site
https://publications-cnrc.canada.ca/fra/droits
LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

National Research Council Canada | Conseil national de recherches Canada

Canada

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC · CNRC

A System for the Seamless Integration of Personal Messages using Agents Developed on a Lotus Notes Platform *

Liscano, R., Impey, R., Gordon, P., and Abu-Hakima, A.
November 1996

* published in the Proceedings of CASCON 96: Meeting of Minds. Toronto, Ontario, Canada. November 12-14, 1996. NRC-40150.

Copyright 1996 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Canada

A System for the Seamless Integration of Personal Messages using Agents Developed on a Lotus Notes Platform.

Ramiro Liscano Roger Impey Paul Gordon Suhayya Abu-Hakima

Abstract

We introduce a seamless messaging system for the management of personal messages. The aim of it is to intercept, filter, interpret, and deliver multi-modal messages (voice, fax, and/or e-mail messages). Messages are delivered to the recipient regardless of their target messaging device. Seamless messaging involves finding the person (if urgent) and delivering the information to them on their cellular phone, pager, laptop, nearest fax, telephone, or workstation. The system includes a set of personal agents that classify and act on incoming messages based on their content. The user specifies the classes and actions to the agent as a set of high-level rules. This allows the user to specify rules that are independent of the messaging system and target devices. A personal agent "Secretary" is responsible for mediating between the different messaging environments, the target devices, and other interacting applications (e.g., calendars, e-mail programs, etc.).
The design of this Seamless Messaging System is based on the integration of three technologies, ubiquitous computing, information filtering, and telematics. The system has been implemented on a Lotus Notes platform. What makes the system unique is its approach to treating a message in a universal manner, its ability to mediate between different messaging devices, and its ability to try to determine the availability of the user.

1. Introduction

In 1991, Mark Weiser [1] published an article that defined a vision of the next generation of computation. He referred to this model as Ubiquitous Computing. As described by Weiser, Ubiquitous Computing can be characterized by two main attributes; ubiquity and transparency. Ubiquity is achieved when interactions are not channeled through a single workstation, therefore access to information is "everywhere." For example, a person should be able to access or receive notification of personal messages from any device available to them (i.e. cellular phones, pagers, laptops, fax machines, telephones, or workstations). In this environment wireless networks will be essential in supporting mobile and remote access. Transparency is the attribute that reflects how well the technology is integrated into the general ecology of the home or work environment. This domain of human device interfacing shall not be addressed in this article. The devices selected for seamless messaging coincide with those that are prevelent in business transactions, telephones, fax machines, and workstations.
In 1994, the Ontario Telepresence Project [2] came to an end and one result of this research was a new model for telematics [3] which addressed the problems of human to human and human to computer interaction by dividing the context of interaction into those that occur in the foreground and those that occur in the background. Foreground human-human and human-computer interactions require high bandwidth and have been exploited by the use of telephones and graphical user interfaces. Contrary to this, human-human and human-computer interactions in the background are low bandwidth persistent operations that have not been investigated very much. In particular the use of the computer to monitor its environment and act as a proxy to the human has not been exploited. This is due primarily to absence of input sensors interfacing with the computer and the difficulties associated with interpreting any resultant sensory data. The area of telematics is key to the addition of enhanced intelligence and background computing. Adding telematics to current computing environments will likely help the user in the management of his/her interactions with other people.
Today's intense communication environments require users to interact across heterogeneous networks and applications. As these communication barriers across networks come down the amount of information readily available increases dramatically and it becomes necessary to filter that information. This paper addresses the problems associated with the management of personal me

ssages across heterogeneous networks. This field of study we call "Seamless Messaging" (SM) primarily involves the integration of the following key technologies, ubiquitous computing, information filtering, and telematics. Most systems [4,5] that use the term "Seamless Messaging" primarily addresses the integration of incoming messages from different communication sources and the accessibility of these messages across heterogeneous networks. They combine very simple message filtering and little telematics into their systems.

Seamless Messaging permits the creation, encoding, filtering, and delivery of messages across heterogeneous networks. The SM paradigm allows the recipient of the message be located if necessary and the message tailored to the recipient's active target device. SM allows users to work in distributed personal workspaces and have messages created and delivered how, when, where, and if they wish them to be. To manage the large number of messages sent to a user, it is also necessary to filter through the messages to reduce the user's workload. In short, its purpose is to provide a dynamic multi-modal receipt, management and delivery system for a user's messages.

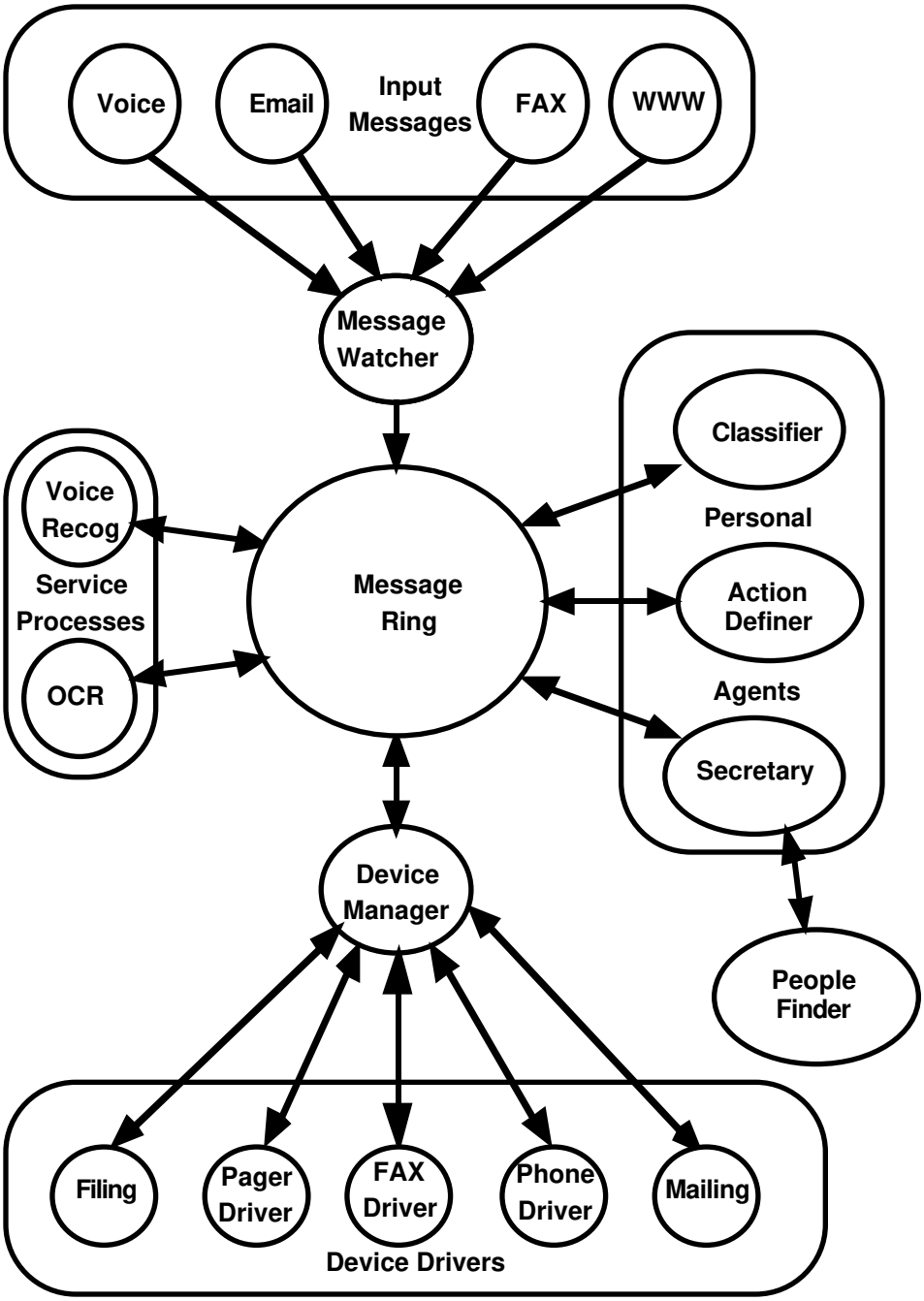


Figure 1. System Architecture

2. Modeling Seamless Messaging

In developing the architecture for the messaging system, there are three main considerations: modularity, facility of modification and generality of the processes.
The modularity of each of the system's components facilitates their development because it helps isolate problems and more easily allows task specialization (while minimizing concerns). In terms of usage, it is best for the system to be modular so that process inter-dependency and single-instance problems would be more easily handled. This modularity also provides the flexibility to change the system architecture without reworking all of the components. This is essential because the addition of new components allows the system to be more comprehensive.
The generality of the central processes is also essential because of the multiple formats of message input and output that are to be supported. Since this integration of multiple media is a key feature of the system, generality of the processes was emphasized. Figure 1 shows the structure of the SM system and the flow of information between the processes of the system. Notice the symmetry between input and output: the former is of many types and must be viewed the same way, the latter is of one type, but viewable many ways.

2.1 System Modules

The system modules or processes can be divided into 4 groups of processes: the "Message Watcher", the "Personal Agents", the "Device Manager", and the "Service Processes".

2.1.1 Message Watcher

The "Message Watcher" was designed as a monitoring process that waits for incoming messages and formats these messages into a unified representation for the system to process. It is the first process and is triggered by an incoming message.

2.1.2 Personal Agents

A Personal Agent is a software process designed to act on behalf of the user by performing a particular task. This is a rather open definition and can encompass a large variety of different types of processes. All personal agents share one common design criteria; they are designed to perform a particular well defined task based on a set of constraints specified by the user and must be customizable by the user at run time. They alleviate the need for the user to perform particular tasks by performing these tasks themselves. This particular system contains three personal agents: a "Classifier", an "Action Definer", and a "Secretary" agent.

2.1.2.1 Classifier Agent

The "Classifier" agent's role is to look through an incoming message and classify the message into a set of classes as defined by the user. The user can define any number and any type of classes for a message along with a set of rules that helps determine if the message fits the defined class type. These rules take on an appearance similar to the following rules,

DEFINE class_name AS
((attribute [operator] value)
OR (attribute [operator] value).....

AND ((attribute [operator] value)
OR (attribute [operator] value)...)

AND ((attribute [operator] value)
OR (attribute [operator] value)...)
. . .

In this example, the class_name will be associated with the messages that evaluate the expression after the keyword AS to TRUE. This expression is composed of a number of (attribute [operator] value) sets connected by Boolean operators. The currently available attributes are sender, recipients, number of recipients, date received, number of lines, subject, message body, and other defined classes. The operators available are =, !=, <=, >=, <, >. Obviously not all are applicable to all fields, and the operator meaning can differ. For example, "number of recipients = 2" means an equality, whereas "body = 'this is an example'" means "body contains 'this is an example'".

2.1.2.2 Action Definer Agent

The "Action Definer" agent's role is to take a classified message and attach to that message an action to be performed on the message. Specification of the type of action to be performed to the message is similar to that done for the classifier agent i.e. as a set of rules that map classes of messages to actions. A typical example of a rule has the following form,

IF class_name ACTION (action_1, action_2, ..., action_n).

Which specifies that if a message has been classified as class_name perform actions (action_1, action_2, ..., action_n).

Actions are slightly more complicated than classes and are divided into 2 types: simple and compound actions. Simple actions are actions that can be resolved directly by the "Device Manager" and

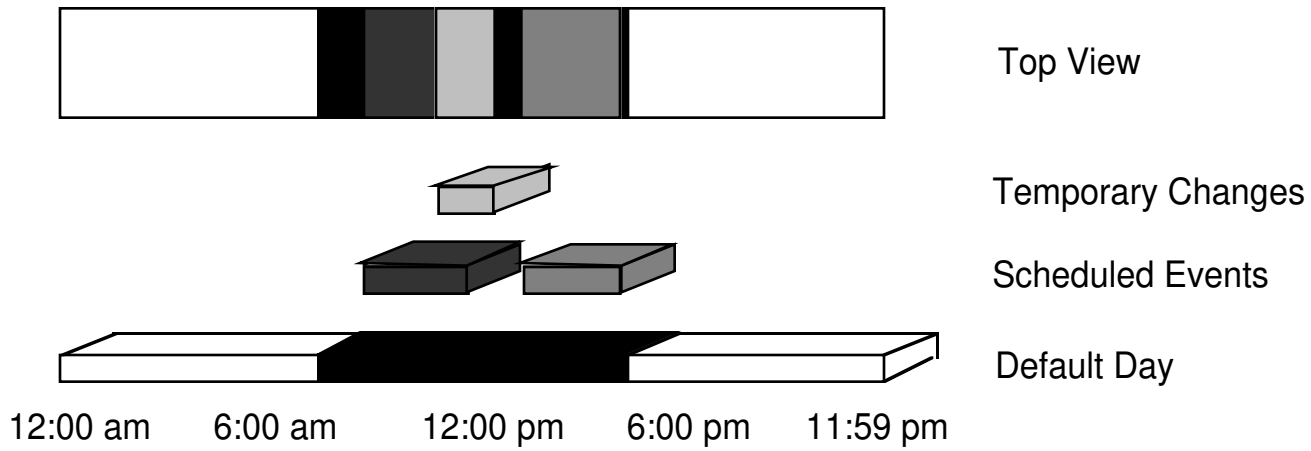


Figure 2. Hierarchical time-management scheduler

require little or no interpretation. These actions can be serviced directly by the "Device Manager", see section 2.1.3. Compound actions are not as device specific as simple actions and require more interpretation and therefore are processed by the "Secretary" before being serviced by the "Device Manager". Examples of these two types of actions are the following, "phone_me '555-1234'" is a typical simple action, while "contact_me" is a compound action requiring further interpretation. These more abstract compound actions are open to interpretation and depend on the particular process and the situation of the user or available inputs.

2.1.2.3 Secretary Agent

The "Secretary" agent's role is to maintain the state of the SM system and act upon any of the actions attached to a message. The secretary must first decide the type of action the message has, is it a simple or a compound action. If it is a simple action then it can be passed directly to the "Device Manager" so it can process the message. In case of failures it is the "Secretary's" role to try an alternative or report a failure. The "Secretary" agent maintains a log of the transactions on the messages. If the message has the more complex compound action then the "Secretary" may have to spawn a helper agent to interpret the action. An example of a helper agent is the "People Finder" agent shown in Figure 1, which is used to resolve the contact me compound action.

2.1.2.4 People Finder Agent

The "People Finder" agent uses whatever resources are at its disposal to determine where the user may be. At the moment it uses a unique hierarchical scheduling structure to determine the availability of the user. The scheduling concept uses a 3 tier hierarchical scheduler that represents 3 abstractions of time management. Commencing from a default description of a user's day (9-5 Workday), to a more refined blocked-off scheduled events (10-12 talk, 2-4 meeting), to finally a set of temporary changes which reflect the instant picture of where the user is located (11-12:30 John's Office). The more detailed temporary changes take precedence over the user's scheduled events which in turn take precedence over the default day. This idea is what is shown in the Top View of the hierarchical scheduler shown in Figure 2.

2.1.3 Device Manager

The "Device Manager" attempts to deliver the message using the appropriate device. It does this by invoking the device driver and passing it the required information. The device driver may need to massage the body of the message into an appropriate format for the physical device. For example, if the message is text only and the target device is only able to handle audio, (i.e. a telephone), some text to speech conversion must be carried out. If the device is limited in the amount of text it can handle, (i.e. a pager device), the device driver must select the limited data to transmit. Many other restrictions and limitations may exist, which the device driver must find a means to deal with in a way that hides the details from the other processes, (i.e. the user is only concerned with generic device names like telephones, faxes, pagers, etc, and not with any particular details of formatting the message for the devices themselves). In this manner the personal agents in the system use a standard message format which may or may not conform to the target device's format. If the device driver is unsuccessful in delivering a message, an error message is returned to the "Device Manager", who must decide if the error should be forwarded to the "Secretary" agent or it can be

resolved by the "Device Manager". For example, if the "Device Manager" has been unsuccessful in the delivery of a message, the "Secretary" agent may be able to re-examine the message and its associated information to determine some additional course of action.

2.2 Workflow

The "Message Ring" is used to symbolize the fact that the processes communicate with each other by placing inter-process message blocks into a common area that the other processes have access to. Message blocks contain an identification tag which triggers the appropriate recipient process to take the message out of the common area and interpret it. The system is easily extendable by defining a new process and the type of messages the process should respond to. Although conceptually any process can communicate with another process on the message ring, in reality the workflow is predetermined.

The system is triggered by some message arriving and this message can be any one of several types, (e.g., Email, FAX, Voice mail, etc). The initiation of the workflow is performed by the "Message Watcher" process which formats an input message with the required header information and places a message block on the "Message Ring" together with the identification tag that will indicate the next operation to be performed on the message. Normally, the identification tag will trigger the "Classifier" agent to read the message block, and using the header information, retrieve the required information about the message. The "Classifier" agent formats a new message block which it places on the Message Ring indicating the message has been successfully classified. The next step is taken by the "Action Definer" which defines the action to be performed based on the class of the message. Again, a new message block is created by the "Action Definer" and placed on the ring. This time the "Secretary" agent is triggered whose role is to interpret and log any of the actions attached to a message. There will be some situations where the "Secretary" may not have sufficient information to process the message, for example when the message is of an urgent type and the user must be located. In this case additional helper agents, like the "People Finder" in Figure 1, may be required to help find the user. Once the "Secretary" determines what device is to handle the message a message block is created and placed on the "Message Ring" to trigger the "Device Manager" to determines how to use the device to contact the user.

3. Implementation

The SM system was implemented in a Lotus Notes 3.x (LN) environment [6]. LN was seen to contain many native features
that would facilitate a messaging system development - good data management across computers and networks, a setup geared towards workflow, multiple format data handling, and ease of database development. At a very early stage in the system's development cycle it was determined that reliance on internal LN processing, i.e. macros and formulas, would restrict the system's key chracteristics of modularity and fast moving workflow. There were many constraints on the functionality of LN macros that made it difficult to access and share information in a realistic amount of time. For this reason the processes discussed in section 2 have been implemented as LN Server Tasks that use the "C" language API to access Notes databases.

3.1 Lotus Notes Databases

A set of Lotus Notes databases were developed to facilitate the sharing of information among the processes as well as being the conduit for specifying the user's desires to the Personal Agents. Figure 3 is representative of how the processes interact with the LN databases. The direction of the arrows signify the flow of data among the databases and the processes.

3.1.1 Message Box

The Message Box database contains the messages that the user has received. At the time of writing this paper only e-mail messages are contained within it. Currently it is the standard notes mail database, but the intention is to use a custom mail database that inherits from the standard mail database with the addition of new fields that will improve the systems performance (e.g keywords associated with the message that can be used for a quick keyword-based classification).

3.1.2 Class Definitions

The "Class Definitions" database is used by the user to specify a set of classification rules, similar to those described in section 2.1.2.1, that personalize the "Classifier" to the the user's needs. The following set of rules represent some typical class definitions of "Messages from Home", List Messages", "Talk Announcements", and "Urgent" classification.

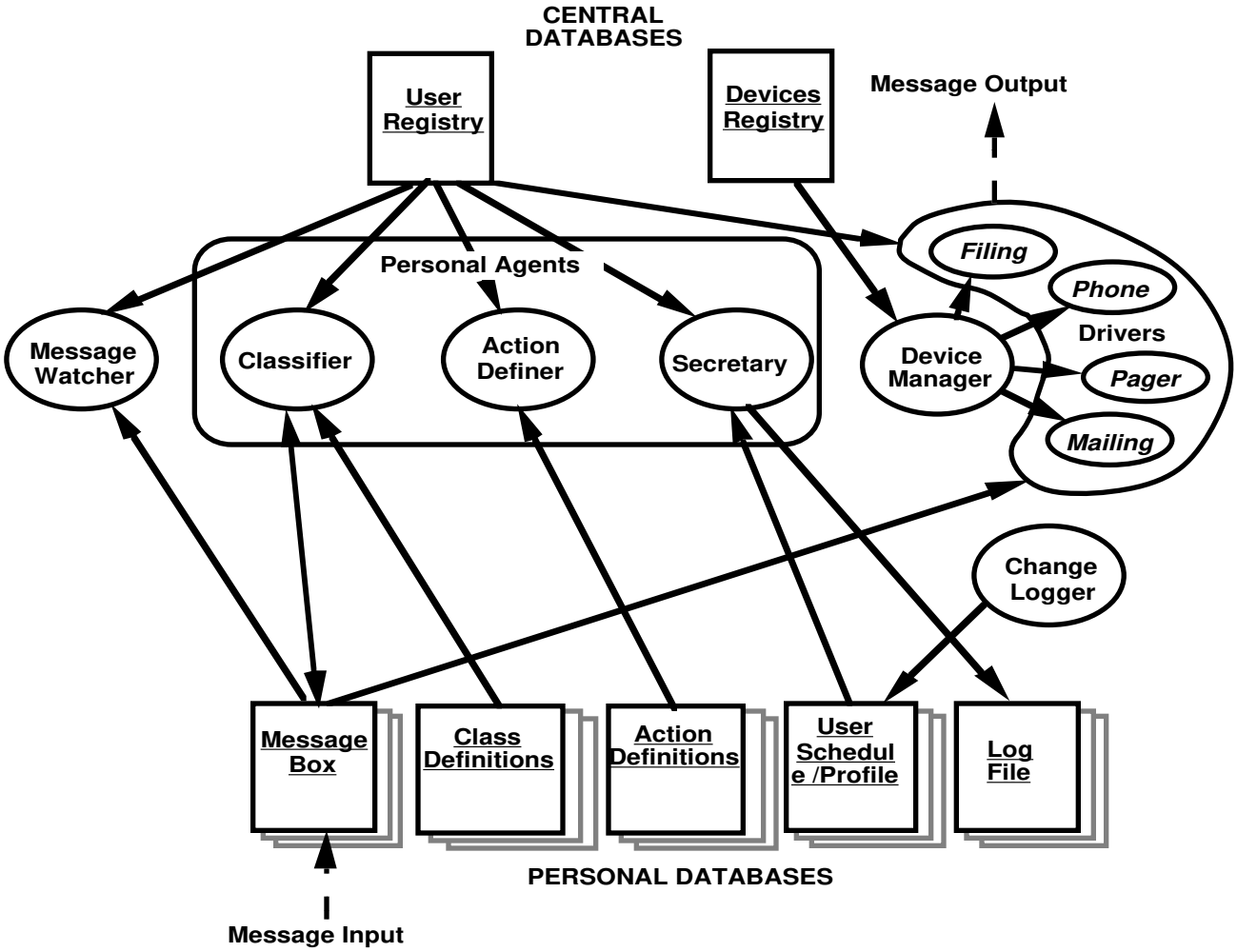


Figure 3. Processes and the Lotus Notes Databases.

```
DEFINE Messages from Home AS ( FROM
        = Mary Smith )
DEFINE List Messages AS ( TO > 5 )
DEFINE Talk Announcements AS (
        KEYPHRASE = seminar )
        OR ( KEYPHRASE = announcing )
        OR ( KEYPHRASE = talk )
DEFINE Urgent AS (Subject = Urgent )
        OR ( CLASS = Messages from
        Home)
```

In these examples any message that contains "Mary Smith" in the "From" field will be classified as "Messages from Home", "List Messages" are those that have 6 or more recipients, "Talk Announcements" are those messages that contain any of the following phrases "seminar", "announcing", or "talk", and "Urgent" messages contain the phrase "Urgent" in the subject or are messages already classified as "Messages from Home". It is important to point out that all messages, regardless of their source, have a standard appearance to them so that at the classification operation no difference is made between whether Mary Smith's message was a telephone call, voice mail, e-mail, or fax.

3.1.3 Action Definitions

The "Action Definitions" database is used to specify a set of mappings from classification names to actions and are a set of rules, similar to those described in section 2.1.2.2. Some typical types of action rules are,

```
IF Urgent ACTION ( contact me )
IF List Messages ACTION ( file
        "listmsgs.mail" )
IF Talk Announce ACTION ( notify me )
```

These examples show the use of the 2 types of actions available, simple and compound actions. The actions "contact me" and "notify me" are compound actions which cannot be resolved directly by the "Device Manager" and therefore require to be serviced by the "Secretary". The action "file" is a simple action which is easily resolved by the device driver "Filing".

3.1.4 User Schedule/Profile

The "User Schedule/Profile" is a calendaring database used by the "People Finder" agent to determine where the availability of the user and the devices that may be used to access that person. The "User Schedule/Profile" database allows the user to record their scheduled events and default day time allocations. It also maintains the temporary changes to the scheduler but the interface to these is through the process "Change Logger" which is a PhoneNotes application that services incoming telephone calls. This is done using straight forward keypad input, prompted by a recorded voice.

Time slots in the schedule have specific device and access key combination information in them, or an availability. Such availabilities are user shorthands, so that if one specifies frequently to use a telephone with access 555-1234, one might create an availability called "At Home" which maps to that. The following are examples of availability shorthands,

At Home	telephone	555-1234
At Office	fax	555-4321
Busy	file	mailmsgs.mail

Notice that these shorthand mappings are really mappings to simple actions that the device drivers can interpret.

3.1.5 Device Registry

Similar to the compound action specification in section 3.1.4 simple actions and their associated device drivers are specified in the "Device Registry" database in the following manner,

file	filedrv.exe
telephone	phonedrv.exe
page	pagedrv.exe

The "Devices Registry" database is used by the "Device Manager" in section 2.1.3 to specify the mapping between simple actions and device drivers.

3.1.6 User Registry

The "User Registry" database serves two main purposes. Firstly, it is used to determine the messaging status of the user (i.e. on, classify only, off). Secondly, it acts as a reference point for the modules, containing the data of where to find the user's personal information (e.g. Classification Rules). Because of these two functions, it is an essential database for the system, tracking the user's status and allowing the personalization of the generalized processing modules.

3.1.7 Log File

The "Log File" is a way to maintain records of the system's state and history on the incoming messages. Notice that the "Secretary" maintains this "Log File" implying that the "Secretary" must in some manner be an overseer of the performance of the system.

3.2 Workflow in Lotus Notes

Section 2.1 already described in general the flow of information in the system. In this section we will give an overview of how this was implemented using the LN environment. As mentioned previously, it was necessary to develop the system by writing external processes acessing the Notes databases through the "C" API. The "Message Ring" described earlier was implemented as a LN "Server Queue" used by the processes running on the console as server tasks. The communication between the tasks is depicted in Figure 4.

.

The text items outlined in dark in the Event Queue are the tags on the message blocks. Each of the server tasks creates message blocks with a particular tag. When the message block is placed on the server queue by its creator, the task that processes message blocks with that particular tag will pick up that message block and process it. For example, the "Classfier" triggers off the events tagged "NEW" which are placed on the queue by the "Message Watcher". The flow of information follows a clockwise pattern in Figure 4 starting from a "Message Input", to the "Message Watcher", "Classifier", "Action Definer", "Secretary", "Device Manager" and finally out through a "Device Driver".

4. Discussion and Future Work

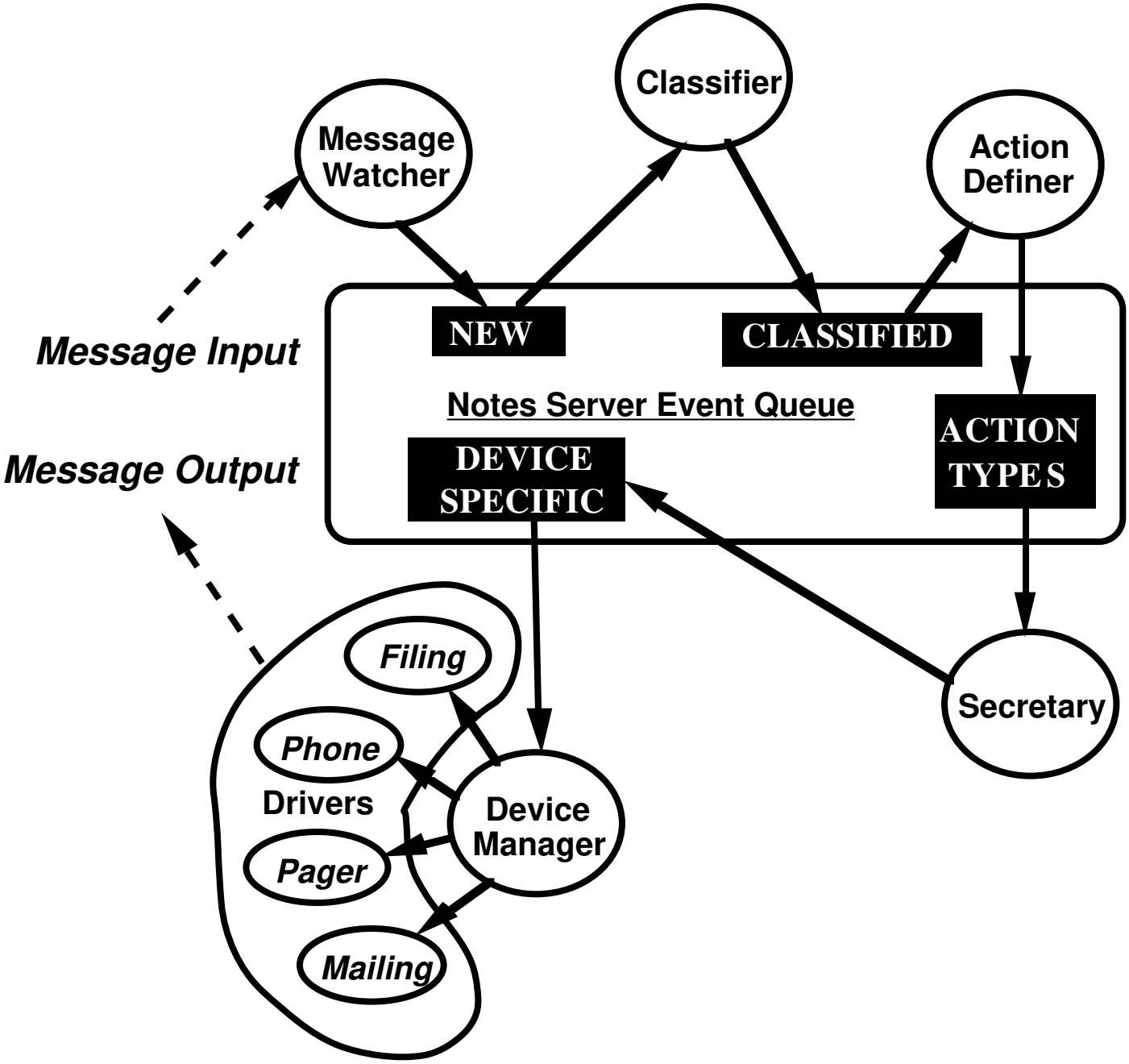
We have described a system for the management of personal messages of multiple mode formats, that is, voice, fax or e-mail messages. A subset of this system has been implemented using Lotus Notes as the underlying development system.
Lotus Notes offered the following benefits:

1) proved to be a stable and flexible platform when running on the OS/2 "Warp" operating system,
2) took care of database access from multiple processes and running the various server tasks and agents in a timely manner,
3) provided many of the network and communications needs of the system; for example, delivery of e-mail, fax and pager gateways.

Using COmmercial off The Shelf components (COTS) reduced the time needed to develop the system, and, more importantly, greatly increased it reliability and functionally. For example, PhoneNotes made the interface between the Messaging system and the telephone very straight forward.

The system, as currently implemented, will enable us to investigate how the system can be used by a group of co-workers. The first such group will be the Seamless Personal Information Networking (SPIN) at NRC. We feel it is important to use the system on a daily basis in order to evaluate its strength and weaknesses. The Personal Agents are evolving as the system is used in our daily work. Learning how to manage the Class and Action rule databases will require some time. The Classifier and associated filter process will need to be improved and adapted as we learn more about how we ourselves manage our messages. The current implementation uses a deterministic regular expression procedure written in PERL [7]. We are actively working on a new implementation of the filter process, one which uses machine learning. This easy replacement of the various modular processes and tasks is one of the major advantage of the architecture we have adopted.

A significant amount of work remains to be done. This version of the system is a prototype that allows us to experiment further either with the modules of the system or the design of the system itself. The system as currently implemented is geared towards asynchronous interactions, we have started work on



extending the system to handle synchronous types of communications. The most important type is interactive voice, that is, telephone calls. We hope to fully integrate interactive telephone handling into the system next. This will include voice mail. We also hope to experiment with delivery of both interactive voice and voice mail to a multi-media equipped workstation. The will involve forwarding voice objects in the appropriate format to the workstation in the case of voice mail, and, somewhat more difficult, interfacing speech right to the desktop via a Computer Telephony Integration (CTI) application. We also hope to carry out work in the area of background monitoring for telematics. Background monitoring involves using various sensors, for example, motion detectors or video cameras, to gain information about the user environment, which can be used by the "People Finder"[8].

About the authors

Ramiro Liscano is a Research Officer with the Seamless Personal Information Networking (SPIN) group of the National Research Council. His current research work is in Reasoning with Uncertainty, Multi-Agent System Architectures and Information Filtering. Before joining the SPIN group in April 1995, he was part of the Autonomous Systems Laboratory at NRC working on Sensory Data Processing for Mobile Robot Navigation.

He can be reached at the Institute for Information Technology, National Research Council Canada, Montreal Road, Building M-50, Ottawa, Ontario K1A 0R6.

His Internet address is liscano@iit.nrc.ca

Roger Impey is a Research Officer with the Seamless Personal Information Networking (SPIN) group of the National Research Council. His current research work is Seamless Messaging. Before joining the SPIN group in April 1995, he was part of the Software Engineering Laboratory at NRC.

He can be reached at the Institute for Information Technology, National Research Council Canada, Montreal Road, Building M-50, Ottawa, Ontario K1A 0R6.

His Internet address is impey@iit.nrc.ca

Paul Gordon is a Co-op student with the Seamless Personal Information Networking (SPIN) group of the National Research Council. He spent the Jan-May '96 term working with the Seamless Messaging team.

He can be reached at The Department of Mathematics, Statistics and Computing Science, Dalhousie University, Halifax, Nova Scotia, Canada.

His Internet address is pgordon@ug.cs.dal.ca

Suhayya Abu-Hakima is the Group Leader of the Seamless Personal Information Networking (SPIN) group at the Institute of Information Technology at the National Research Council. She has been a researcher at NRC since 1987. She had previously worked at BNR starting in 1982. She has Doctorate and Masters degrees in AI from Carleton University's Department of Systems and Computer Engineering.

She can be reached at the Institute for Information Technology, National Research Council Canada, Montreal Road, Building M-50, Ottawa, Ontario K1A 0R6.

Her Internet address is Abu-Hakima@iit.nrc.ca

References

[1] M. Weiser, "The Computer for the 21st Century", Scientific American, Vol. 265, No. 3, 1991, pp. 94-104.

[2] J. Chattoe, P. Leach, and R. Riesenbach, "Ontario Telepresence Project - Final Report", Information Technology Research Centre, Telecommunications Research Institute of Ontario, March 1995.

[3] R. Buxton, "Integrating the Periphery and Context: a New Model of Telematics", *Graphics Interface '95* , Morgan-Kaufman, New York, 1995, pp. 239-246.

[4] CallXpress3 Product Information Sheet, Applied Voice Technology, Kirkland, Washington, 1996.

[5] The OcteLink Network Service Product Information Sheet, Octel Communications, Milpitas, California, 1996.

[6] P. Gordon, "Message Handling Application", NRC COOP Work Term Report, 1996.

[7] L. Wall and R. L. Schwartz, "Programming PERL", O'Reilly & Associates, Sebastopol, CA, 1990.

[8] Ferguson, Innes A. and Davlouros, James D. 1995. PeopleFinder: A Multimodal Multimedia Communications Tool for Interconnecting Network Users. In Working Notes of the IJCAI-95 Workshop on Intelligent Multimedia Information Retrieval, Montreal, PQ, August