

A Call Handling Assistant for Mobile Devices

Wayne Wobcke, Rita Chan and Andrew Limaru
School of Computer Science and Engineering
University of New South Wales
Sydney NSW 2052, Australia
{wobcke|ychan|alimaru}@cse.unsw.edu.au

Abstract

With the near ubiquity of mobile phones, people are reachable almost anywhere and at any time. The convenience this provides for people to more effectively combine different aspects of their lives has the consequence of an increased blurring of boundaries, for example between work and home. Thus, as one sphere impinges upon others, we see a need for people to limit their interactions on the mobile phone, to control (within socially acceptable bounds), when they are reachable and by whom. In this paper, we present a call handling personal assistant that runs on mobile devices to enable users to manage calls, based on the caller and caller's group, the time and day, and their activity. The system uses a new rule-based technique, Prioritized Ripple Down Rules, which gives the user a high level of confidence in the system's behaviour.

1. Motivation

Mobile phones are nearing ubiquity in the general population of the Western world. This means people can reach, and can be reached by, anyone almost anywhere and at any time. This has led to immense changes in the way people manage their time and social interactions with others. The mobile phone means people can work remotely, can be reached in cases of emergency, and in turn, can reach others when they need to. However, this has also led to a blurring of boundaries between different social spheres, e.g. between work and home. We believe people now perceive a need to limit their social interactions on the mobile phone, to preserve some personal time and space when they are not constantly reachable by just anyone. Thus we see a need for people to personalize their mobile phones, automatically being able to accept or reject calls, have the phone ring or not ring (e.g. in public places) or to automatically send a reply (via SMS) when they do not wish to be contacted.

Recent studies of mobile phone use in Japan have identified a number of trends related to current usage in that country, which most likely foreshadow or mimic trends elsewhere. First, Matsuda [7, 8] identifies 'selective sociality', the tendency of young people to interact with a range of acquaintances based on a variety of social contexts, however limiting the majority of their mobile phone interaction (calls and e-mail) to 5–10 very close friends, or their 'full-time intimate community'. Second, there are noticeable gender differences, Matsuda [8], Dobashi [2], e.g. women tend to have more interaction with friends and neighbours whereas men with work colleagues (it is argued that this is a reflection of their existing personal networks). Third, young people have their own patterns of mobile phone use, Miyaki [9], in which mobile phones play a role for young people to 'define a sense of group membership', Ling and Yttri [5]. All these studies suggest that mobile phones should be highly flexible and personalized, and based on an individual's social relationships and contexts.

In this paper, we present a personal assistant intended for use on a mobile phone that allows people more control over how calls are handled, i.e. accepted or rejected, whether the phone rings or not, and whether to automatically send an SMS reply. The system enables users to manage incoming calls based on the caller and caller's group, the time and day, and the user's activity. The main technical issue addressed, as is usual with personal assistants, e.g. Maes [6], is that the system should not overly burden the user in its requirement to construct and maintain a user model representing the user's call handling preferences. Moreover, the user interface of our system, which currently runs in simulation on a PDA platform, must adhere to the limitations of mobile devices in terms of screen size, processing power, memory and bandwidth, making usability paramount. In addition, a high degree of accuracy is required of the system, e.g. a guarantee that certain important calls would never be missed is essential; more generally, users must have confidence that the assistant will work as anticipated.

In our earlier work on e-mail management, Ho,

Wobcke and Compton [3], we took a rule-based approach to user modelling, in particular based on Ripple Down Rules (RDR), Compton and Jansen [1]. Ripple Down Rule systems are suitable for personal assistants because they allow the incremental construction of a user model, are comprehensible by the user, and provide a high level of accuracy. However, while there are similarities between phone call handling and e-mail management, there are also differences, most particularly the limited information about phone calls available to make a decision (with e-mail, the assistant has access to the content of the message), making accuracy an even greater concern. In this paper, we present a variant of RDR, called Prioritized Ripple Down Rules, that is appropriate for the call handling domain.

2. Prioritized Ripple Down Rules

Ripple Down Rules (RDR), Compton and Jansen [1], are systems of ‘if-then’ rules organized in a hierarchy of rules and exceptions (exceptions to rules may themselves have exceptions, etc.). RDR systems are incremental; starting with a rule base containing only a default rule at the root of the tree, the user repeatedly adds or refines rules (creates exceptions to existing rules) to correctly classify a sequence of examples. This process results in a high degree of accuracy since, whenever the rule base is modified, the RDR engine ensures that all previous examples continue to be classified correctly.

There are various types of RDR system; with Single Classification RDR (SCRDR), the rule base defines one conclusion for any given example, whereas with Multiple Classification RDR (MCRDR), Kang, Compton and Preston [4], there is a set of such conclusions. Each conclusion for an example is found by following a path in the RDR tree to the most specific node whose rule is applicable to the example (i.e. the example satisfies the rule’s conditions) and returning the rule’s conclusion.

An example MCRDR rule base for the call handling domain is shown in Figure 1 (in section 3, we describe a scenario showing how this rule tree is constructed using the call handling assistant). In this tree, the rules are named *Rule 0*, *Rule 1*, etc., in the order of their creation. The conditions of the rules are constraints on the caller and call group, the time and date, and the user’s current activity; the conclusion is a “profile” to be used in handling the call, specifying whether to accept or reject the call, and if the call is accepted, whether to allow the phone to ring or not (silent mode), and whether to automatically send an SMS reply or not. The default rule, *Rule 0*, is that any condition gives the *Default* profile, which is under the control of the user. The five rules underneath *Rule 0* are rules defining what to do in various situations, e.g. *Rule 1* states that if the user’s activ-

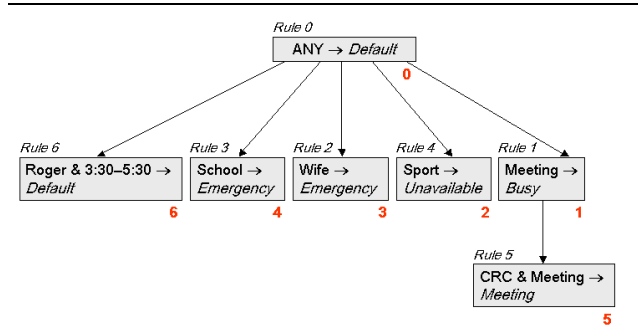


Figure 1. Prioritized Ripple Down Rule Base

ity is ‘Meeting’ to use the *Busy* profile. *Rule 5* is an exception to this rule; if someone from the CRC calls while in a meeting, use the *Meeting* profile instead. So, for example, if someone from the school calls while the user is in a meeting, both *Rule 1* and *Rule 3* are most specific rules that apply to this example, and the MCRDR rule base returns the set of conclusions {*Emergency*, *Busy*}.

A key feature of standard RDR systems is that the system provides extensive help to the user in building the rule base, so much so that the user never has to examine the rules themselves in order to resolve conflicts. Rather, a standard RDR system makes use of “cornerstone cases”, those examples that triggered the formation of a rule, in order to help the user resolve potential conflicts as new rules are defined. When a new rule is being defined, in the context of a new example, the RDR system presents those cornerstone cases that would conflict with the new rule, allowing the user to adjust the rule’s conditions, if necessary, *before* committing to the new rule. In addition, when defining a new rule to correct an incorrect conclusion, the system is able to add all necessary refinements to the rule base at once.

However, in the call handling application, a more “rule-oriented” approach is needed, since users do not deal with sets of examples (calls), though they can use the call history as a repository of examples. Moreover, as the system must produce determinate behaviour for each call, a single classification is needed, suggesting the use of Single Classification RDR. But the rule systems for SCRDR typically become more complex than those for MCRDR (they effectively have only binary branching), so for the call handling assistant, we have chosen to use Multiple Classification RDR with a prioritization mechanism to ensure that only one classification is given for any example (that of the rule with highest priority applying to the example). The priorities define a total ordering on the rules in the RDR tree, and are shown in Figure 1 below each node. There is only one constraint on the priority ordering: any refinement rule must have a higher priority than its parent (so the default rule has the lowest priority of all); otherwise, the priority

ordering is defined by the user, and the user interface must provide an intuitive interface for manipulating priorities as well as for defining rules. With these priorities, if someone from the school calls while the user is in a meeting, *Rule 3* alone will be used, so the *Emergency* profile will be used to handle the call.

3. Call Handling Assistant

A call handling assistant based on Prioritized Ripple Down Rules has been implemented in Java (Jeode Personal Java 1.1.8) on a PDA (HP iPaq hx4700) using hsqldb as the database engine. While it would have been desirable to implement the assistant on a mobile phone platform, at the time of development, the hx4700 had a faster processor, more memory and a larger (4 inch) screen with four times the resolution of the closest comparable mobile phone.

The system provides the basic functions supplied with a mobile phone, though cannot be used to make calls (calls are simulated through a wireless network). The graphical user interface is organized as a series of tabs: Rules, Profiles, Phonebook, Calendar and History, as shown in Figure 2(a). The Phonebook allows the user to set up a list of contacts and call groups (with individuals possibly belonging to multiple groups, as is natural with Multiple Classification RDR). The Profiles tab enables the user to define a range of actions to be taken when receiving a call (accept/reject, ring/silent, and send SMS), and there is a *Default* profile which the user can modify. The Calendar tab allows the user to define a list of activities with associated days and times, and the History tab stores a list of calls received. The Rules tab is the most important, as it provides both a mechanism for manipulating rules (adding, refining and prioritizing), and for searching (displaying a subset of the rules that match a particular set of conditions). The search function was anticipated to be useful when users had a large enough rule set, e.g. that would not fit on one screen (around 9–10 rules). Furthermore, the subset of rules resulting from a search can be prioritized with respect to one another and the system will maintain a single consistent priority ordering. This provides a convenient way to prioritize a larger rule set by focusing on those rules that could potentially apply in a given situation or range of situations.

The current mode of the assistant is displayed in the blue title bar, and for Rules is generally one of *search* or *edit*. The default search mode is to display those rules matching the user's current activity and the current day and time. These are the current active rules, which are the rules that are applicable in the current situation, giving the user some idea of the range of possible behaviour of the assistant at that time. The rules are displayed in priority order (highest priority at the top). Note, however, that the hierarchical structure of the MCRDR rule base cannot be displayed using this method,

and users need a separate mechanism to infer this structure (an error message is displayed when they attempt to redefine the priority of a rule to be lower than that of its parent in the rule tree). When a new rule is added, the system displays the set of rules that are compatible with the new rule (similar to how potential conflicts are shown in standard RDR), being the set of those rules that can be meaningfully prioritized with respect to the new rule. The heading above the list of rules changes to indicate when prioritization of rules is possible. The user's current activity and the current time are displayed in the pale blue bar at the top of the screen below the title bar. The activity is taken from the user's calendar, though it can also be manually adjusted.

The following scenario illustrates the main features of the call handling assistant. We assume the user has already set up some profiles (*Busy* and *Emergency*), and defined three rules (if activity is Meeting, use *Busy* profile; if Wife calls, use *Emergency* profile; and if School calls, use *Emergency* profile). Searching with ANY in all fields lists all of these rules.

To modify the system's behaviour in a new situation, the user needs to define a new activity (Sport), profile (*Unavailable*) and then a rule (if activity is Sport, use *Unavailable* profile). The assistant displays all compatible rules, and the user prioritizes the new rule (using the green arrow buttons) to be just above the default rule, which always has lowest priority, as shown in Figure 2(a).

Suppose sometime later the user receives a call while in a meeting. The 'Meeting' rule applies, hence the *Busy* profile is used, so the call is rejected. The user now wishes to correct this behaviour by defining an exception to the 'Meeting' rule that applies when the caller is from the CRC group. This is done by selecting the rule and pressing the *refine* (R) button. The rule's existing conditions are greyed out (these cannot be modified) and the user selects more specific conditions. In this case, the user selects the CRC group and defines a new *Meeting* profile (not to be confused with the Meeting activity) to accept the call in silent mode and send a reply via SMS. This rule is given the highest priority.

However, the user realizes that an important call from Roger in the CRC is expected soon whilst possibly in a meeting. So the user defines a new "ad hoc" rule for Roger calling between 3:30 and 5:30 (this is not a refinement of the 'Meeting' rule and it applies even when the user is not in a meeting). This rule is the new highest priority rule, as shown in Figure 2(b), and will be deleted automatically when it is no longer applicable, i.e. at 5:30.

Finally, at 6:00 the user searches the rule set for rules that apply for either CRC callers or during a meeting; the default rule and both the 'Meeting' rule and its exception for CRC callers are displayed, as shown in Figure 2(c). In search mode, the assistant acts more like a search engine and returns all rules that match any one of the conditions.

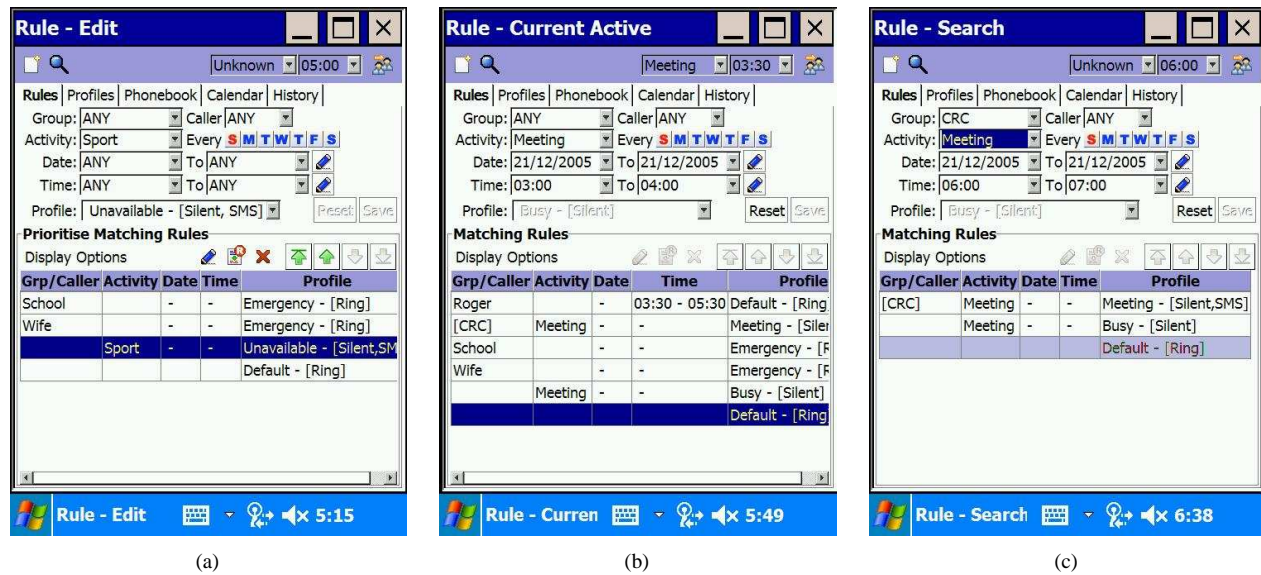


Figure 2. PDA Displays for Sample Scenario

4. Conclusion and Further Work

With the prevalence of mobile phones, there is a need for people to more effectively control the nature of their social interactions. In this paper, we presented a personal assistant application for call handling, focusing on the issue of how users can specify the intended behaviour of their phone. The call handling assistant uses a new rule-based technique, Prioritized Ripple Down Rules, that ensures a high degree of accuracy, and which therefore engenders the confidence of users in the system's behaviour. Rules can be defined in terms of the caller and caller's group, the current day and time, and the user's activity, and determine a profile to be used in handling the call, whether to accept or reject the call, whether to ring or not, and whether to automatically send an SMS reply. This provides a highly expressive language for defining the call handling assistant's behaviour. In future work, we plan to incorporate location awareness into the system so that users can define rules referring to their current location, and to undertake a usability study of the system.

Acknowledgements

This work was funded by the CRC for Smart Internet Technology. We would like to thank the members of the Secure Identity Management project, especially John Zic and Christine Satchell, for discussions of their work on the topic of identity management on mobile devices. Thanks also to Ibrahim Bokharouss and Van Ho for their contributions to the project.

References

- [1] P. J. Compton and R. Jansen. A Philosophical Basis for Knowledge Acquisition. *Knowledge Acquisition*, 2:241–257, 1990.
- [2] S. Dobashi. The Gendered Use of *Keitai* in Domestic Contexts. In M. Ito, D. Okabe and M. Matsuda, editors, *Personal, Portable, Pedestrian*. MIT Press, Cambridge, MA, 2005.
- [3] V. H. Ho, W. R. Wobcke and P. J. Compton. EMMA: An E-Mail Management Assistant. In *Proceedings of the 2003 IEEE/WIC International Conference on Intelligent Agent Technology*, pages 67–74, 2003.
- [4] B.-H. Kang, P. J. Compton and P. Preston. Multiple Classification Ripple Down Rules: Evaluation and Possibilities. In *Proceedings of the 9th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, pages 17.1–17.20, 1995.
- [5] R. Ling and B. Yttri. Hyper-Coordination via Mobile Phones in Norway. In J. E. Katz and M. A. Aakhus, editors, *Perpetual Contact: Mobile Communication, Private Talk, Public Performance*. Cambridge University Press, Cambridge, 2002.
- [6] P. Maes. Agents that Reduce Work and Information Overload. *Communications of the ACM*, 37(7):31–40, 1994.
- [7] M. Matsuda. Discourses of *Keitai* in Japan. In M. Ito, D. Okabe and M. Matsuda, editors, *Personal, Portable, Pedestrian*. MIT Press, Cambridge, MA, 2005.
- [8] M. Matsuda. Mobile Communication and Selective Sociality. In M. Ito, D. Okabe and M. Matsuda, editors, *Personal, Portable, Pedestrian*. MIT Press, Cambridge, MA, 2005.
- [9] Y. Miyaki. *Keitai* Use among Japanese Elementary and Junior High School Students. In M. Ito, D. Okabe and M. Matsuda, editors, *Personal, Portable, Pedestrian*. MIT Press, Cambridge, MA, 2005.