METHOD AND APPARATUS FOR MODELLING USER BEHAVIOUR

Embodyments of the invention are concerned with providing a method of modelling user behaviour for use in predicting user actions in respect of an event, where a user action is one of a set of actions. An event may be, for example, receipt of a phone call from a caller, in which case the user action is one of answer call, or divert call to voicemail. The method comprises the steps of: receiving current context data comprising at least data relating to the received communication; performing, in dependence on said current context data, a plurality of processes each corresponding to an assumed user characteristic and each producing an estimate of probability of an action; weighting the probability estimates in accordance with stored factors, each indicative of the previous success of the respective process in correctly predicting an action; identifying, from said probability estimates, a predicted user response. Thus rather than using an attribute/value representation, a user model representation is used to predict user actions.
METHOD AND APPARATUS FOR MODELLING USER BEHAVIOUR

The present invention relates a method of modelling user behaviour, and is particularly suited to modelling behaviour that has a set of well-defined user actions associated therewith.

In the field of user modelling, a significant amount of research and development has been directed towards improving the way in which information is identified for, and presented to, users. In particular, user modelling is concerned with the way in which content of information is identified to be of interest to a user; how information (of interest) is delivered to a user; and the way in which interface devices interact with a user when providing the user with information. More recently, user modelling has additionally concentrated on adapting user models in response to changing user behaviour.

The applications of user modelling are vast, and include identifying and conveying information and goods of interest to a user; identifying and conveying entertainment, such as television programmes and music, of interest to a user; and identifying areas where users need assistance. User modelling has also been applied to machine co-ordination; for example, the latest Volkswagen Beetle™ uses fuzzy logic control programs, which constantly monitor the current speed and use of the accelerator pedal, in its automatic gearbox. These programs classify the driver with a degree of sportiness, and adjust the car’s gear-changing behaviour accordingly.

Several techniques have been developed to improve ways in which information is identified to be of interest to a user. These include user profiling (enabling information content to be personalised to suit the user’s interests) and clustering (collaborative filtering based on likes and dislikes of users).

Concurrently, techniques have been developed to improve interaction between a user and devices. One such technique is the Bayesian model, which attempts to identify a user’s goals based on the user’s observed actions and queries. An example of a product that embodies such a technique is the Microsoft™ Office Assistant™, which aims to provide appropriate help to a user, and to identify and remedy gaps in the user’s knowledge. Another technique is that of expert systems, where an expert model of use of, say, a software package, is derived, and user models (based on observed user behaviour) are compared to the expert model.
Techniques for adapting user models in response to changing user behaviour include machine learning methods, where a user model is typically represented as a set of attribute-value pairs. Machine learning is based on observed data, and includes methods such as rule-based knowledge learning and neural networks. A problem with neural networks is that the network appears to an onlooker as a "black box", so that it is almost impossible to identify any kind of logic, or rule set, that governs the extrapolation behaviour of the neural network. For this reason, rule-based methods are preferred (rules can more easily be reviewed and modified). However, a problem with rule-based methods is that it can be extremely difficult to extract the relational and background knowledge that is required to formulate the rules in the first instance.

An alternative method is a relational learning method known as inductive logic programming, which derives rule-sets from observed data. Typically a significant amount of data is required as input, and the learning process is a formidable search task.

Most of these machine learning methods require a relatively large, static set of training examples to train a model. The incorporation of new training examples normally involves restarting the learning process with a new, expanded, training set. As the learning process is typically quite slow, this is clearly undesirable. Additionally in user modelling it is relatively expensive to gather training data, as explicit feedback is required from the user. A particular example of this type of system is described in International Patent Application PCT/US98/03626, (publication number WO98/38781) which describes use of a Neural Network to identify a user's location as a function of date and time of day. This location is then used to identify sequences of telephone numbers for forwarding calls. The computational overhead associated with a Neural Network means that it is only trained once a day, so that the Neural Network alone cannot be relied on to predict the user's whereabouts. Accordingly when new data is received, the Neural Network generates an output in respect of the new data, and modifies the output by a modification factor. This modification factor is essentially a probability, which, immediately after training, has a value of 0.9, and then decreases throughout the day as a function of time. Alternatively the output from the Neural Network is ignored, and the probability
values alone are used, in conjunction with last recorded telephone numbers for the user.

An alternative approach to modelling user behaviour is described in “Collaborative Interface Agents”, Lashkari, Metral and Maes, in Software Agents 1994, pp. 444 – 449. This method employs so-called learning interface agents to learn a user’s behaviour based on training examples. The interface agents employ Memory Based Reasoning (Standfill and Waltz, 1986, “Towards Memory Based reasoning”, Communications of the ACM 29 (12): 1213-1228) to capture user patterns. Essentially, when a user takes an action in respect of a received communication, that action is paired with aspects of the communication, and recorded in the agent’s memory. This means that the agent can calculate, on the basis of aspects of incoming communications, a likely action of the user, and proactively manage the communication for the user. Lashkari et al identify a particular problem with this method, namely that its performance is directly related to the number of similar training examples available. Thus for new situations (e.g. a new communication having hitherto unseen aspects) the agent recommends an action that is unrepresentative of the user’s true action. Their solution to this problem is to identify, from other user’s, agents that have seen communications similar to the new communication, and recommend an action based on a weighted sum of actions stored by such identified agents.

According to a first aspect of the invention there is provided a method of predicting user response in respect of a received communication, wherein a user response is one of a set of actions, the method comprising the steps of:

receiving current context data comprising at least data relating to the received communication;

performing, in dependence on said current context data, a plurality of processes each corresponding to an assumed user characteristic and each producing an estimate of probability of an action;

weighting the probability estimates in accordance with stored factors, each indicative of the previous success of the respective process in correctly predicting an action;

identifying, from said probability estimates, a predicted user response.
The term "user characteristic" is used interchangeably with the term "user type" in the following description.

These processes can be collectively viewed as a "user model", and each stored factor is representative of the degree to which the process relating thereto has correctly predicted a user response. Conveniently the method includes, for each user characteristic, comparing the predicted user response with actual user response, and adjusting the stored factors in dependence on whether the respective method has produced an estimate which alone is indicative of an action corresponding to the actual user response.

It is instructive to compare the present invention with the system described by Lashkari et al (referenced above), since this document is similarly concerned with predicting user behaviour. In Lashkari et al a closest matching situation is identified on the basis of similarity between aspects of an incoming communication and previously analysed communications, and an action corresponding to the identified situation is retrieved. In other words, the prior art makes decisions (runs processes etc.) in respect of the caller. So, for the situation/action rule system: a call comes in, the system analyses features of the call - e.g. who the caller is, consults a look-up table (or rules) and identifies an action associated with that caller.

By contrast, the method of the present invention takes account of contextual data relating to the user, and evaluates the likelihood, or otherwise, of a user performing a particular action. The output of each process is a probability value, and a user response is predicted on the basis of a weighted combination of the probability values and stored factors.

Preferably the probability estimates are expressed as probability intervals, which are particularly well suited to user modelling, as they remove a need for assumptions relating to the action being modelled.

In some embodiments of the invention there are two or more methods associated with a user characteristic, so that the performing step includes combining output from the said two or more processes in accordance with a predetermined functional relationship so as to generate the further probability estimates.

Conveniently the context data includes data representative of a user's activities, in the form of a schedule or similar, so that at least one of the processes involves determining availability of user during a defined period. In addition, or
alternatively, the context data includes data indicative of the immediacy of an interaction between caller and user, where the interaction can be one, some or all of a meeting or/and email or/and a phone call. Advantageously a communication is a phone call, and the user response is either “answer call” or “divert call to voicemail”.

Preferably the context data includes data relating to telephone calls between the caller and user, and to the length of telephone calls received from caller during a time period. In addition, or alternatively, the method can include determining a number of calls made by the caller to the user during a time period.

Advantageously the processes utilise rules embodied as fuzzy logic expressions.

Embodiments of the invention can additionally be applied to receipt of email messages, scheduling tasks into a diary, and programming of devices.

Conveniently there is provided apparatus corresponding to the afore-described method.

In the context of embodiments of the present invention, a “user” is not necessarily limited to a human entity, as it may include a piece of equipment or some software.

Further aspects and advantages of the present invention will be apparent from the following description of preferred embodiments of the invention, which are given by way of example only and with reference to the accompanying drawings, in which

Figure 1 is a schematic diagram of a general purpose computer operable to store and process embodiments of the invention;

Figure 2 is a schematic diagram showing software components that can interoperate with embodiments of the invention;

Figure 3 is a schematic diagram of components of a response predictor according to embodiments of the invention;

Figure 4 is a flow diagram showing a method of modelling user behaviour according to embodiments of the invention;

Figure 5a is a flow diagram showing aspects of the method of Figure 4;

Figure 5b is a flow diagram showing further aspects of the method of Figure 4;

Figure 5c is a flow diagram showing yet further aspects of the method of Figure 4;
Figure 6 is a schematic diagram showing a functional relationship utilised by
the method of Figure 4;

Figure 7 is a schematic diagram showing a further functional relationship
utilised by the method of Figure 4; and

Figure 8 is a flow diagram showing the ways in which incoming calls are
processed and recorded.

Overview of operating environment for embodiments of the invention

Figure 1 shows a generally conventional computer system H1 that comprises:

a conventional keyboard 101; a display screen 103, such as a CRT or plasma
screen; a mouse 105; a processor (CPU) 107 such as a Pentium ™ processor; a
memory store 109; data storage 111; and input/output interfaces 115 to connect
the workstation to a local area network (LAN) and wider area networks (WAN) such
as the internet, to facilitate data exchange including email messaging with remote
users connected to such networks. The interface 115 also allows control of a plain
old telephone set (POTS) and the components shown in Figure 1 are interconnected
by a common bus 117. In addition to the single system configuration shown in
Figure 1, several computer systems (not shown) may be interconnected over a local
area network via the input/output interface 115.

In a conventional manner, under control of operating system programs 121
(shown in Figure 3), the processor 107 runs programs held on storage 111, making
use of the memory store 109, under the control of keyboard 101 and mouse 105,
together with imaging device 114 to provide data on the display 103. Audio inputs
can be made through the audio input 113 for use by speech recognition software.

Referring also to Figure 2, an intelligent assistant system 219 may be stored
on the storage 111 for processing by the processor 107. The intelligent assistant
system 219 enables users to devote their time to complex tasks while the system
219 takes some decisions on behalf of the user based on previous observations of
the user, thus enabling the user to increase productivity. Typical tasks to be
performed by the system include time, information and communication management.
When the computer system comprises several computer workstations,
interconnected via the input/output interface 115, several intelligent assistant
systems 219 may be active and may communicate with one another.
As shown in Figure 2, such a system 219 may comprise a set of autonomous systems 203, 205, 207, 209, generally referred to as agents or assistants, specialising in various tasks such as diary management, telephone call filtering and email prioritisation, web search and telephone directory enquiry:


System 205 comprises an email assistant for email management;

System 207 comprises a telephone assistant, which determines whether a call should be answered immediately, or whether the call may be postponed to a later time or date.

System 209 comprises a diary assistant, which accepts diary entries entered by a user;

Each agent has its own interface and interacts with the user in its own particular way, and the agents communicate with one another by message passing. These agents are essentially reactive agents that respond to events in the user’s environment (such as emails and telephone calls) and initiate interactions with the user.

As stated above, these agents 203, 205, 207, 209 perform tasks on behalf of the user. This means that at least some of the agents will have to perform some degree of user modelling in order to behave in a manner that typifies user actions and retrieves information that is relevant to the user.

In particular, with one of the known methods described above, prediction of user response to, say, an incoming call, could involve the telephone assistant 207 consulting a user profile. A user profile defines attribute/value relationships, so that the telephone assistant 207 would typically firstly determine the attribute (caller), then identify a corresponding value from the user profile, and finally perform an action based on the identified value. For example, if the caller is identified to be the user’s wife, the value (action) could be “answer”, whereas if the caller is identified to be someone from the administration department at work, the value (action) could be “send to voicemail”.
A problem with this approach is that it does not take any account of the user's state of mind or of current activities. It may be that the user is trying to complete a report, for which there is a deadline of 6:00 pm that evening; in this situation, the user is unlikely to want to be interrupted by a call from anyone.

User profiles could of course be modified to include filters such as "if the user is busy, divert the call", but this incurs additional processing and will only work if there is a filter to match every possible activity that the user may be engaged in; furthermore, user profiles cannot attempt to factor in the state of mind of the user.

Embodiments of the present invention move away from attribute/value representations and instead utilise a user model to predict user actions, as is described in more detail below. In general, embodiments are suitable for predicting one of a well-defined set of user actions, such as, in the case of a user receiving a telephone call, either answering the call, or not answering the call.

Overview of embodiments of the invention

A first embodiment of the invention can be used to predict user response to incoming telephone calls. Essentially the first embodiment involves training a user model using data representative of previously received phone calls. The data includes at least caller details, the context of user at the time of receiving the call, and the way in which the user handled the call (i.e. answer, divert to voicemail).

Training of the model has the objective of replicating observed user characteristics so that the model can thereafter be used to predict user response, and thus manage incoming phone calls. The user model comprises a plurality of programs representative of user characteristics – such as busy, talkative, antisocial etc. Each of the characteristics has a "behaviour" associated therewith – e.g. if a user is deemed to have the characteristic "antisocial", then this is coupled with a behaviour of "never answers calls".

Training a user model involves inputting data thereto and, for each of the calls, determining likely behaviour of the user in respect of the call. Firstly the identity of the caller is determined. Secondly the context of the user, at the time of the call, is determined – e.g. from consulting the user's diary it can be determined that the user was in a meeting, or was free at the time of the call; and thirdly the caller identity and user context is input to the user model (specifically to programs
representative of the user characteristics mentioned above), to determine likely behaviours of the user.

Once the likely behaviours have been determined, they are used to identify whether, on a balance of probabilities, the user would have answered, or otherwise, the call. As will be described below, this is compared with how the user actually handled that call, and used to update the user model.

Once the model has been trained it can be continually modified, using real-time data.

Advantages of the invention arise from the way in which the user model operates: as the model can change each time a new observation is made, it does not require lengthy re-training phases. This is an improvement over most of the so-called machine learning systems, which require an initial training phase and are then static unless complex retraining takes place. In addition, and as a consequence, the model effectively adapts to reflect the user's changing, or unchanging, method of responding to telephone calls.

Referring to Figure 3, the first embodiment will be described in more detail. Generally embodiments of the invention are referred to as a response predictor 300, and comprise at least some of programs 311, 313, 321, 322, 323, 325, 331, 333. Preferably the response predictor 300 forms part of the telephone assistant 207, and as such is stored on storage 111, and is processable by the processor 107. The skilled person will appreciate that each program can comprise a suite of programs.

The programs include a user model program 311, which comprises a model UM of the user, together with plurality of characterising programs 321, 322, 323, 325 each of which determines a behaviour of the user.

The programs additionally include a caller model program 313, which comprises models CM of each caller (i), together with plurality of caller programs 331, 333 each of which maintains data relating to calls received from callers.

At least some of the characterising programs 321, 322, 323, 325 request and receive information from various sources, such as a caller database ST (shown in Figure 2), and the diary assistant 209, and each of the characterising programs 321, 322, 323, 325 generates, as output, a value indicative of a probability of characteristic having a bearing on the way in which the user will handle a call.
For example, one of the characterising programs 321 identifies whether the user can be characterised as talkative, another 323 whether the user can be characterised as busy. For the talkative characterising program 321, the program 321 has a static output of "always answers". For the busy program 323, the program 323 identifies, based on information gathered from the various sources, whether the user is free, or busy, and outputs a value representative of the degree to which the user's "busy-ness" will have a bearing on the user's response to an incoming call.

The store ST includes information (such as caller details and duration of calls) corresponding to the previously received calls.

The user model program 311 receives, from each characterising program, output indicative of behaviour corresponding thereto, and predicts a user response based on the output. Subsequently the user model program 311 updates the user model UM in accordance with how closely the predicted user response matched the actual behaviour recorded for that call, which is also available from the store ST.

At least some of the caller programs 331, 333 request and receive information from various sources, such as the caller database ST, and maintain a record of e.g. call types, duration and frequency, in order to generate an output indicative of how well a caller conforms to a caller type.

For example, one of the caller programs 331 identifies whether the caller can be described as brief, another 333 whether the user can be described as frequent. For the brief caller program 331, the program 331 has an output of "always makes short calls to the user". For the frequent caller program 333, the program 333 identifies, based on information gathered from the store ST, whether the caller calls the user on a regular basis, and outputs a value indicative of the likelihood that the caller is a frequent caller.

The operation of the response predictor 300, when training the user model UM according to this first embodiment of the invention, will now be described with reference to the flowcharts shown in Figures 4, 5a, 5b and 5c.

User Model

Training of the user model UM involves use of call data that has already been seen and stored in the store ST, or equivalent, such as a log file (not shown).
Typically training is based on calls received over a few days, taking account of the user's diary record over those few days (e.g. by accessing information maintained by the diary assistant 209).

The user model UM is most easily described by way of example. As stated above, a user model program 311 includes programs, each representative of a user characteristic, some of which have been mentioned above, but which generally include at least talkative, antisocial, busy, overloaded, interactive, regular, selective and prioritising.

Examples of some of the user characteristics and behaviours corresponding thereto are defined in Table 1:

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
<th>Data required</th>
<th>Behaviour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Talkative</td>
<td>User prefers to answer all calls</td>
<td>None – static</td>
<td>Always answers</td>
</tr>
<tr>
<td>Antisocial</td>
<td>User doesn't like to answer calls</td>
<td>None – static</td>
<td>Never answers</td>
</tr>
<tr>
<td>Busy</td>
<td>User has small proportion of free time in working day</td>
<td>Diary entries during time period T (configurable); Information relating to caller</td>
<td>Answer if call is likely to be short</td>
</tr>
<tr>
<td>Overloaded</td>
<td>User has small proportion of free time in working day</td>
<td>Diary entries during time period T (configurable)</td>
<td>Never answers</td>
</tr>
<tr>
<td>Interactive</td>
<td>User has had a recent call or meeting with caller</td>
<td>Diary entries during time period T (configurable); call log</td>
<td>Answers</td>
</tr>
<tr>
<td>Regular</td>
<td>User handles a large proportion of calls at this time of day</td>
<td>Call log</td>
<td>Answers</td>
</tr>
</tbody>
</table>

Table 1

The user model includes a value, (or factor), for each of the characteristics, which indicates a measure of the number of times that the behaviour of the user has conformed to that characteristic. E.g. suppose the current user model UM is:

talkative (UM_talkative): 0.7
antisocial (UM_antisocial): 0.3
busy (UM_busy): 0.8
interactive (UM_interactive) : 0.6

This says that in 70% of cases so far, behaviour has conformed to the talkative characteristic (i.e. the call has been answered), in 30% of cases behaviour has conformed to the antisocial characteristic (not answered). (NB this is a probability distribution because these two are mutually exclusive and exhaustive i.e. one or the other is always true, never both).

In cases where the busy characteristic is applicable, this says that the user model UM correctly predicts user behaviour 80% of the time (the busy characteristic is only applicable when there is a small proportion of free time in a specified period (as described below)). In cases where the interactive characteristic is applicable, this says that the user model UM correctly predicts user behaviour 60% of the time (the interactive characteristic is only applicable when the user has recently been in contact with the caller (as described below)).

The user model UM is updated by the user model program 311, which, at step S 4.1 retrieves information relating to a call. This includes retrieving time of the call, together with the identity of the caller from the store ST.

At step S 4.2 each of the characterising programs 321, 322, 323, 325 etc is processed. At least some of the characterising programs request information representative of diary information and caller details, and these programs process the information in accordance with internal routines. The operation of some of the characterising programs is shown in Figures 5a, 5b and 5c – for the talkative, busy and interactive characterising programs 321, 322, 325, 327.

Referring to Figure 5a, at step 501 the programs 321, 322 are activated and the talkative program returns "answer" while the antisocial program 322 returns "divert to voicemail". Each of these programs returns a value that indicates that they are equally relevant (1.0).

The busy characteristic program 323 is then processed. Referring to Figure 5b, at step 511, the busy characterising program 323 receives data representative of the user’s activities and the identity of the caller, and generates a value that is indicative of the relevance of the busy characteristic to prediction of a user response. The busy characteristic 323 is coupled with user response "divert to
voicemail", so that, in the event that the busy characteristic is deemed relevant to a
user's response, higher values indicate that the user is less likely to answer the call.

Accordingly busy characterising program 323 requests part of the user's diary
corresponding to a working period (which is configurable and may, for example, be
defined as working hours in a 24 hour period surrounding the time of the call), and
receives it. Next, at step 513, the busy characterising program 323 determines the
amount of free time that was available to the user during the working period (V1).

At step 515 the busy characterising program 323 translates this amount of
free time into a relative measure by dividing number of unoccupied working hours by
available working hours and, at step 517 compares this with a previously determined
definition of "free time". E.g. assuming the busy characterising program 323
calculates the amount of free time, V1, to be 13% and the definition of a "small
amount of free time" is having less than 10-15 % (fuzzy number) free time, such a
value (13%) constitutes a small amount of free time.

As stated above, the busy characterising program 323 only generates an
output in cases where there is a small amount of free time. If the amount of free
time exceeds the fuzzy definition of a "small amount of free time" (here 10-15%),
the user is not deemed to have been busy, and the busy characterising program 323
exits without generating an output. If this is the case, the busy characteristic will not
be taken into account when predicting user response at step S 4.3 (described
below).

As a value of 13% does fall within the fuzzy definition of a "small amount of
free time", for this example, the busy characterising program 323 proceeds to the
next step 519, where the busy characterising program 323 determines the type of
caller for the current caller (caller A). Preferably this includes accessing information
stored by the caller model CM, in particular information available from the caller
programs 331, 333.

Information relating to length of calls is maintained by a caller model CM
(described below), so that at step 519 the busy characterising program 323 sends a
request to the caller model CM, with an identification of the caller (caller A), for a
measure of the brevity of phone calls between caller A and the user.

At step 521 the caller model CM sends a value, V2, representative of the
proportion of calls received from caller A that are short, to the busy characterising
program 323; for the present example, assume that a value of 1.0 is sent (V2 = 1.0).

At step 523 the busy characterising program 323 transforms V1 (13%) into a likelihood that the user will answer the phone, given the amount of free time available to the user. Preferably this transformation involves use of a fuzzy function, such as that shown in Figure 6, from which it can be derived that 13% free time maps to 0.4% probability of this characteristic having a bearing on the user’s tendency to answer the phone.

At step 525 the busy characterising program 323 combines the probability value derived at step 523 with the value 1.0 returned from the brief caller program 331 so as to further factor in the brevity of the caller. This gives a probability of 0.4 x 1.0 = 0.4 that the busy characteristic is relevant to the prediction of user response (and can be interpreted as 40% chance that the call will be diverted to voicemail).

At step 527 the busy characterising program 323 outputs the value calculated at step 525 (0.4) to the user model program 311.

Referring to Figure 5c, the interactive characterising program 325 receives an input representative of the identity of the caller, and generates a value that is indicative of the relevance of the interactive characteristic to prediction of a user response. Accordingly, at step 531, the interactive characterising program 325 requests, from the store ST, data representative of the number of calls received from the caller in, say, a 24 hour period preceding the current call. In addition the interactive characterising program 325 requests and receives part of the user’s diary corresponding to a working period (e.g. 24 hour period, but this is configurable). At step 533 the interactive characterising program 325 works out whether the user has had a recent meeting and/or call with the caller.

For example, when the interactive characterising program 325 receives information relating to the user’s schedule, it identifies the time of last meeting, and calculates the time since that meeting. If the meeting was more than, say, 24 hours ago, the user is not deemed to have had a recent meeting with the user, and the interactive characterising program 325 exits without generating an output. If this is the case, the interactive characteristic will not be taken into account when predicting user response at step S 4.3 (described below). Note that this period of time is
configurable, and could be expected to vary depending upon the context within which the invention is working.

However, if the last meeting was less than 24 hours ago, the interactive characterising program 325 calculates, at step 535, a value that is indicative of the relevance of the interactive characteristic to a prediction of a user response, taking into account the immediacy of the meeting. This calculated value is essentially a measure of the likelihood that the user will answer the call, given the immediacy of the last interaction with the caller (so that, for example, if the meeting was within the last hour, there is a 98% chance that the user will answer the call, and if the meeting was 20 hours ago, there is a 4% chance that the user will answer the phone). At step 537 the interactive characterising program 325 outputs the value calculated at step 535.

As an alternative, if the user has both had a recent meeting, and received one or more phone calls from the caller during the 24 hour period, the value output from the interactive characterising program 325 can be higher than if the user has either had a recent meeting with, or taken at least one call from, the caller.

In the case of the current example, it is assumed that the user has not had a meeting with caller A within the last 24 hours, so that the interactive characteristic is not deemed to have a bearing on the inclination of the user to answer the phone.

Once all of the characterising programs 321, 322, 323, 325 have been processed, the user model program 311 predicts a user response, and updates the user model UM in accordance with the correlation between actual and predicted user response.

Thus at step S 4.3 the user model program 311 combines the outputs received from the characteristic programs 321, 322, 323, 325: e.g. say the user model program 311 receives the following outputs (example values):

from the talkative program 321: applicability: 1
from the antisocial program 322: applicability: 1
from the busy program 323: applicability: 0.4
(interactive characteristic is not deemed relevant)
In one arrangement (first approach), the user model program 311 calculates the probability of the user performing one of two actions – answer or divert to voicemail – by performing a weighted sum of the contributions to the two actions. Recalling the User model UM values (stored factors, or confidence values) presented above:

- talkative (UM_talkative) : 0.7
- antisocial (UM_antisocial) : 0.3
- busy (UM_busy) : 0.8
- interactive (UM_interactive) : 0.6

\[
\begin{align*}
\text{answer} &= 1.0 \times \text{UM_talkative} \\
&= 0.7 \\
\text{voicemail} &= \text{Av} [1.0 \times \text{UM_antisocial} + 0.4 \times \text{UM_busy}] \\
&= \text{Av} [0.3 + 0.8 \times 0.4] \\
&= 0.31
\end{align*}
\]

Which gives, as output, a weighting towards action "answer".

Alternatively (second approach) the user model program 311 selects whichever characteristic is the most likely to have a bearing on the user’s response, and chooses an action associated with selected characteristic. For example:

- Talkative characteristic: \(1 \times \text{UM_talkative} = 0.7\)
- Antisocial characteristic: \(1 \times \text{UM_antisocial} = 0.3\)
- Busy characteristic: \(0.4 \times \text{UM_busy} = 0.32\)

So that the action associated with the talkative characteristic (answer) is selected.

At step S 4.4 the user model program 311 retrieves information from the store ST indicative of the actual user response in respect of this call. If the action derived at step S 4.3 concords with the actual user response, then the user model program 311 subsequently updates the user model UM (steps S 4.5 and S 4.6).

At step S 4.5 the user model program 311 updates the user model, which essentially involves, for each characteristic, updating the performance of its ability to
predict the user’s behaviour. The performance is indicative of the number of times the user behaviour has conformed to the behaviour predicted by the characteristic.

In the case of the talkative characteristic, a value of 0.7 (UM_talkative = 0.7) indicates that, of 100 cases, the user has answered the phone 70 times. If, for the 101st case, the user does not answer the phone, the performance of the talkative characteristic is updated to 70/101 (UM_talkative = 70/101).

Similarly, in the case of the busy characteristic, a value of 0.8 (UM_busy = 0.8) indicates that, of 100 cases in which the busy characteristic has been relevant, the user has not answered the phone. If, for the 101th case in which the busy characteristic is deemed relevant, the user does answer the phone, the performance of the busy characteristic is updated to 81/101 (UM_busy = 81/101).

**Caller Model**

As stated above, the caller model program 313 maintains a caller model CM for each of the callers from which the user receives calls, and provides input to the busy characterising program 323.

Essentially a caller model CM comprises programs, each representative of a caller characteristic, some of which have been mentioned above, but which generally include at least brief, verbose, frequent, reactive, proactive. Examples of caller characteristics and descriptions corresponding thereto are given in Table 2:

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brief</td>
<td>always makes short calls to user</td>
</tr>
<tr>
<td>Verbose</td>
<td>always makes long calls to user</td>
</tr>
<tr>
<td>Frequent</td>
<td>calls user frequently</td>
</tr>
<tr>
<td>Reactive</td>
<td>calls following a recent voicemail left by user</td>
</tr>
<tr>
<td>Proactive</td>
<td>calls prior to a meeting with user</td>
</tr>
</tbody>
</table>

**Table 2**

In general, each of the caller programs 331, 333 etc. records statistics relating to calls – e.g. brief caller program 331 records a proportion of “short” calls made from the caller to the user, and frequent caller program 333 records a proportion of calls made over a defined period.
In each of these programs, fuzzy terms are used to define the respective conditions: “short”, “long”, “frequently”, “recent” and “prior”. These are internally defined within each of the programs, and are configurable. For example “short” and “long” calls may be defined using the fuzzy function shown in Figure 7, and “frequently” can be defined using a fuzzy function (not shown) defined as {0:0, 1:0, 2:0.5, 3:0.8, ...}, which states that receipt of 0, 1, 2, 3 calls are represented by the following values: 0, 0, 0.5 and 0.8 respectively.

Operation of these caller programs is most easily explained by an example: assume that calls from caller A (referred to in busy characterising program 323 described above) are recorded in the call log ST as follows:

<table>
<thead>
<tr>
<th>Caller</th>
<th>Length min</th>
<th>direction</th>
<th>action</th>
<th>day</th>
<th>month</th>
<th>start hr</th>
<th>start min</th>
<th>start sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2.1</td>
<td>in</td>
<td>answer</td>
<td>23</td>
<td>7</td>
<td>9</td>
<td>16</td>
<td>44</td>
</tr>
<tr>
<td>A</td>
<td>1.1</td>
<td>in</td>
<td>answer</td>
<td>23</td>
<td>7</td>
<td>15</td>
<td>16</td>
<td>22</td>
</tr>
<tr>
<td>A</td>
<td>0.9</td>
<td>in</td>
<td>answer</td>
<td>23</td>
<td>7</td>
<td>15</td>
<td>22</td>
<td>00</td>
</tr>
</tbody>
</table>

Table 3
The data from Table 3 is input to the fuzzy function shown in Figure 7, and the fuzzy relationship defined above for “frequency” to determine, respectively, a probability that the caller is brief caller and a probability that the caller is a frequent caller. Using this data, the caller model would be:

Brief 1.0 i.e. all calls have been short
Verbose 0.0 i.e. no calls have been long
Frequent 0.8 i.e. probability that caller is a frequent caller is 0.8

(If the lengths of calls vary between 2 and 8 minutes, and can thus be defined as a mixture of “short” and “long” calls, an average call length can be calculated and used to identify a probability of the caller being brief, or verbose from Figure 7).

Clearly the period set to assess length of calls – “minutes” in Figure 7 - could be varied, to suit the environment being modelled.

Once the caller model CM has been trained for this caller, and is in use for real data, the caller model CM is updated using the said real data. Preferably such
real data is stored in the store ST, from which it is accessed by the caller programs 331, 333 etc (this is described below, with reference to Figure 8).

If a call is received from a new caller, a caller model CM corresponding to the new caller is created and initialised to:

5 Brief (0 1) (i.e. probability that the caller makes short calls lies between 0 and 1)
Verbose(0 1)
Frequent (0 1)

and is subsequently modified using data corresponding to calls received from the new caller as described above for Caller A. Note that this makes use of probability intervals rather than point values; this representation is described in more detail below.

Once the user model UM has been trained it can be used to pre-emptively determine whether the user should answer an incoming call, in which case it allows
the user’s telephone to ring; or whether to divert the incoming call, in which case it
diverts the call to voicemail. Alternatively it can allow all incoming calls to ring and offer a suggestion to the user. The suggestion is either that the user should answer the call, or leave the call, in which case the call will be diverted to voicemail. The benefit of the latter arrangement is that the user model UM can continually be
updated using this data – i.e. the user model UM can offer a suggestion, which is a prediction within the meaning of step S 4.3 in Figure 4, and compare that suggestion with the actual user behaviour, which is actual behaviour within the meaning of step S 4.4 in Figure 4.

25 Additional details and modifications

The call database ST is populated with information relating to call duration, caller ID, and actions taken in respect of the call. Caller ID is typically derived from
the Call Line Identifier (CLI) of the incoming call. The call database ST preferably
stores a mapping between CLI and user details (including, e.g. phone numbers for
home, work, mobile etc.), which enables a caller to be identified using the CLI.

Figure 8 is a flow diagram showing possible ways in which a user can receive
an incoming call; as can be seen in the Figure, information relating to calls is stored
in the call database ST after the call has ended.
Other embodiments

The response predictor 300 described above could be modified to predict how a user would handle incoming emails, how a user would schedule, e.g. meetings in his diary, and how a user would filter Short Message Service (SMSs) messages. In addition, the response predictor 300 can be modified to perform automatic video recording of programmes that are regularly viewed by a user.

When used to predict how a user would handle incoming emails, the response predictor 300 could be used to train a user model UM to filter emails as a function of priority: thus into categories read now, read later, read never. The user model program 311 could use previous email data to train the user model UM, as described above, but whereas for the first embodiment the response predictor 300 can be trained to generate one of two actions (answer, do not answer), with email, there may be (at least) three categories (read now, later (itself comprising two categorise such as within 5 days, within 4 weeks), never).

New callers

As stated above, when a user receives a call from a new user, the caller model program 313 can create a new caller model CM for that caller, and initialise all of the caller programs to zero. As an alternative, and if information about the new caller is available, (e.g. interests and/or habits that correlate with telephone preferences), the caller model program 313 could perform clustering between this information and similar information stored relating to existing callers, to identify existing callers that are similar to the new caller. The caller model program 313 could then create a partial caller model CM from that of the identified existing caller(s).

Information gathering

In the above examples, the busy and interactive characterising programs 323, 325 request information relating to the user’s schedule from the diary assistant 209 (or equivalent), and these characterising programs 323, 325 use that information to calculate, e.g. amount of free time available to the user. Alternatively, the diary assistant 209 could be responsible for calculating the amount of free time available
to the user, and could simply return a value indicative thereof to a respective characterising program.

As a further alternative, rather than the characterising programs communicating directly with the diary assistant 209 (or equivalent), embodiments could include an information-gathering program (not shown) whose job it is to periodically collect schedule information from various information sources – e.g. the diary assistant 209. In such an arrangement, the characterising programs would request information directly from the information-gathering program. This method would be particularly useful if information were to be gathered from additional sources such as e.g. GPS – for position and movement data etc., where synchronising of data feeds may be more conveniently handled by a central program.

*General calculation details*

In the example of the user model UM given above, the user model UM utilises point values to describe how well a characteristic represents a user’s behaviour (talkative: 0.7; antisocial: 0.3; busy: 0.8; interactive: 0.6). As shown above in the caller model CM example, the user model UM could alternatively use intervals. This removes a need for assumptions, and implicitly factors in uncertainty, which is particularly useful for user modelling.

The use of intervals can be exemplified by the following example: if an even number is showing on a die, there is zero probability of the number being 1, 3 or 5, and the probability of the number being 2 (or 4 or 6) is somewhere in the interval 0-1, subject to the condition that \( \Pr(2) + \Pr(4) + \Pr(6) = 1 \). Alternatively, this can be expressed by saying that the probability of being in the set \{2,4,6\} is 1.

If using point-valued probabilities, the probabilities would be expressed, respectively, as \( \Pr(2) = 1/3 \), \( \Pr(4) = 1/3 \), \( \Pr(6) = 1/3 \). Expressing probabilities using the point-value method assumes that there is a fair die i.e. that there are equal prior probabilities.

If the probability is expressed over interval values, this reduces the need for prior assumptions. This is particularly useful in modelling user response, which does involve making assumptions and working with a significant amount of uncertainty.

Thus the user model UM described above as talkative : 0.7
antisocial : 0.3
busy : 0.8
interactive : 0.6

could alternatively be expressed as:
talkative : (0.6 0.8)
antisocial : (0.2 0.4)
brusy : (0.75 0.85)
interactive : (0.4 0.8)

When the user model UM is expressed as interval values, prediction of user response (step S 4.3) following the second approach proceeds as follows. Using the values calculated in the example given above:

<table>
<thead>
<tr>
<th>prototype</th>
<th>support i.e. historical success rate of this characteristic</th>
<th>probability that characteristic is applicable at this moment</th>
<th>overall probability of successful prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>talkative</td>
<td>(0.6 0.8)</td>
<td>1</td>
<td>(0.6 0.8)</td>
</tr>
<tr>
<td>antisocial</td>
<td>(0.2 0.4)</td>
<td>1</td>
<td>(0.2 0.4)</td>
</tr>
<tr>
<td>busy</td>
<td>(0.75 0.85)</td>
<td>0.4</td>
<td>(0.3 0.34)</td>
</tr>
</tbody>
</table>

Table 4

The most relevant characteristic is selected (following “approach 2” described above) as follows: in the first instance select the characteristic associated with a highest lower probability bound; if there are two or more intervals with the same lower bound then select whichever of the two has the highest upper bound; and, if there are two or more intervals with the same lower and upper bounds, select from these characteristics at random.

In the current example, the characteristic associated with the highest lower probability bound is “talkative”, which predicts “answer” as a user response.

As described above, when identifying whether the user has a “small amount” of time free, the busy characterising program 323 reviews the user’s diary over a specified period, and derives a discrete amount of time when the user appears to be unoccupied. The busy characterising program 323 then compares this derived discrete data with fuzzy conditions (e.g. “small amount” of free time).
However, this presupposes that diary entries are always "firm" entries – i.e. entries for which the user has a definite engagement. In many cases, a user's diary may comprise at least some tentative diary entries, such as "possible meeting with Boss", so that the busy characterising program 323 is unable to determine the exact amount of free time in the user's diary – i.e. it cannot determine that the user has exactly 13% free time.

In such a situation the busy characterising program 323 determines an approximate amount of free time e.g. "about 13% free time", and includes means, such as semantic unification means, to enable the busy characterising program 323 to process this fuzzy amount of time in the context of fuzzy conditions ("small amount of free time"). Semantic unification unifies fuzzy statements with probability values using mass assignment theory. For more information the reader is referred to: "FRIL - Fuzzy and Evidential Reasoning in AI", Baldwin, J. F., Martin, T. P. and Pilsworth, B. W. (1995), Research Studies Press (John Wiley), ISBN 0 86380 159 5.

Implementation details

The user model program 311 and user model UM described above operates in accordance with Flexible, Incremental Learning of User Models (FILUM) - which is based on dynamic support logic programs, and is described in "Incremental learning of user models – an experimental test bed", T. P. Martin, Proceedings of IPMU, Madrid 2000, 1419 - 1426.

Each of the characteristic programs described above is implemented as a class, and the user model UM is considered to have an (interval-valued) probability of belonging to each class according to how well the class behaviour matches the observed behaviour of the user (as described above, with reference to Figures 4 and 5a, 5b, 5c).

This can be described in general terms by the following equations:

\[ C = \{c_1, c_2, \ldots, c_k\} \], where each \( c_i \) corresponds to a characteristic (i.e. a characterising program 321, 322, 323, 325 etc)

and each class \( c_i \) has a set of possible output values

\[ B = \{b_1, b_2, \ldots, b_n\} \], where \( b_n \) is the probability of the user behaving according to the characteristic.

User model \( m \) predicts the behaviour of a user.
Let $S_n(m \in c_i)$ be the support for the user model $m$ belonging to the $i$th class before the $n$th observation of behaviour. Initially,

$$S_1(m \in c_i) = [0, 1]$$

for all classes $c_i$, corresponding to a state of complete ignorance.

Each time an observation is made, every class makes a prediction, and the support for the user model being a member of that class is updated according to the predictive success of the class:

$$S_{n+1}(m \in c_i) = \frac{n \times S_n(m \in c_i) + S(c_i, \text{Behaviour}_{n+1} = b_{n+1})}{n + 1}$$

where $S(c_i, \text{Behaviour}_{n+1} = b_{n+1})$ represents the support for class $c_i$ predicting the correct behaviour on this iteration.

Thus the algorithm is

```plaintext
S1(m \in c_i) = [0, 1], all i
WHILE another prediction wanted
    m.PredictedBehaviour = m.ResolveConflict([c_i.Behaviour : S(m \in c_i) \times S(c_i.Behaviour), i = 1...k_0])
    update S(m \in c_i) for each i, according to the observed behaviour
ENDWHILE
```

where the ResolveConflict method fuses the supports and predictions from the various classes.

The passing of information between the characterising programs 321, 322, 323, 325 etc and caller programs 331, 333 etc., together with interaction and interface with the diary assistant 209 and call database ST is implemented in the Java programming language. It is understood that use of Java is inessential to embodiments of the invention.

The user model UM and user modelling program 311 may be implemented in Fril (Fril++) but it is understood that this is inessential to embodiments of the invention.
25

CLAIMS

1. A method of predicting user response in respect of a received communication, wherein a user response is one of a set of actions, the method comprising the steps of:
   receiving current context data comprising at least data relating to the received communication;
   performing, in dependence on said current context data, a plurality of processes each corresponding to an assumed user characteristic and each producing an estimate of probability of an action;
   weighting the probability estimates in accordance with stored factors, each indicative of the previous success of the respective process in correctly predicting an action;
   identifying, from said probability estimates, a predicted user response.

2. A method according to claim 1, further including comparing the predicted user response with actual user response, and adjusting the stored factors in dependence on whether the respective process has produced an estimate which alone is indicative of an action corresponding to the actual user response.

3. A method according to claim 1 or claim 2, in which the context data includes data indicative of the origin of the received communication.

4. A method according to claim 3, in which the context data includes data indicative of interactions between the origin of the received communication and the user.

5. A method according to claim 4 in which the interaction is one, some or all of a meeting or/and an email or/and a phone call.

6. A method according to claim 5, in which at least one of the processes involves determining the immediacy of an interaction between the origin of the received communication and the user.
7. A method according to claim 6, in which at least one of the processes involves determining length of telephone calls received from the origin of the received communication during a time period.

8. A method according to claim 6 or claim 7, in which at least one of the processes involves determining a number of calls made by the origin of the received communication to the user during a time period.

9. A method according to any one of claims 5 to 8, in which at least one of the processes involves determining a number of emails sent by the origin of the received communication to the user during a time period.

10. A method according to any one of the preceding claims, in which the context data includes data indicative of the user’s activities during a defined period, and at least one of the processes involves determining the availability of user during the said period.

11. A method according to any one of the preceding claims, in which, when there are two or more methods associated with a process, the performing step includes combining probability estimates from the said two or more processes in accordance with a predetermined functional relationship so as to generate a probability estimate.

12. A method according to any one of the preceding claims, wherein the received communication is a phone call, and the user response is one of answer call, or divert call to voicemail.

13. A method according to any one of claims 1 to 11, in which the received communication is an electronic mail message, and the user response is to prioritise email into one of a plurality of priority categories.
14. A method according to claim 13, in which the user response includes storing email in a storage location representative of one of the plurality of priority categories.

15. A method according to any one of claims 1 to 11, in which the received communication is a request to schedule a task into a schedule, and the user response is to insert the task into the schedule as a function of constraints placed upon the task.

16. A method according to any one of claims 1 to 11, in which the received communication is a broadcast of information, and the user response is to programme a programmable device to record the broadcast information.

17. Apparatus for predicting user response in respect of a received communication, wherein a user response is one of a set of actions, the apparatus comprising:

   receiving means arranged to receive current context data comprising at least data relating to the received communication;

   processing means arranged to perform, in dependence on said current context data, a plurality of processes each corresponding to an assumed user characteristic and each producing an estimate of probability of an action;

   weighting means arranged to weight the probability estimates in accordance with stored factors, each indicative of the previous success of the respective process in correctly predicting an action;

   identifying means arranged to identify, from said probability estimates, a predicted user response.

18. Apparatus according to claim 17, including modifying means arranged to compare the predicted user response with actual user response, and to adjust the stored factors in dependence on whether the respective process has produced an estimate which alone is indicative of an action corresponding to the actual user response.
19. Apparatus according to claim 17 or 18, wherein at least one of the processes is operable to determine immediacy of an interaction between the origin of the received communication and the user.

20. Apparatus according to any one of claims 17 to 19, in which at least one of the processes is operable to determine availability of user during a defined period.

21. A method according to claim 19 or claim 20 in which the interaction is one, some or all of a meeting or/and an email or/and a phone call.

22. Apparatus according to claim 21, wherein at least one process is operable to determine length of telephone calls received from the origin of the received communication during a time period.

23. Apparatus according to claim 21 or claim 22, wherein at least one process is operable to determine length of telephone calls received from the origin of the received communication during a time period.

24. A method according to any one of claims 21 to 23, wherein at least one process is operable to determine a number of calls made by the origin of the received communication to the user during a time period.

25. Apparatus method according to any one of claims 17 to claim 24, wherein at least some of the processes utilise fuzzy logic expressions.

26. Apparatus according to any one of claims 17 to 25, in which the probability data are expressed as intervals of probabilities.
S 4.1 Retrieve information relating to a call

S 4.2 Process characterising Programs 321, 323, 335

S 4.3 Combine outputs received from characteristic programs

S 4.4 Retrieve information indicative of actual user response in respect of this call for each characterising program

S 4.5 Evaluate performance of predicted output and update user model UM

Interactive program 325:

531 Request data representative of diary content and calls received from the caller in previous 24 hours

533 Determine whether user has had recent meeting and/or call with caller

Has user had recent call/meeting with Caller?

NO: Exit

Yes

535 Calculate value indicative of immediacy of meeting/call

537 Output value calculated at step 535

Fig 4

Fig 5c
**Fig 5a**

Busy program 323:

1. Request part of user’s diary corresponding to a working period
2. Determine amount of free time available to the user during the working period (V1)
3. Translate V1 into relative measure
4. Compare V1 with previously determined definition of “free time”
5. **Is V1 small amount of free time?**
   - **Yes**
     - Determine type of caller for the current caller
     - Receive value V2 representative of brevity of calls between caller and user
     - Transform value V1 into value indicative of user answering phone, on basis of free time
     - Combine transformed V1 with V2
     - Output combination of V1 and V2
   - **NO**: Exit

**Fig 5b**

Talkative: Return “Answer”
Antisocial: Return “Divert to voicemail”
Fig 8

Incoming call with caller-ID

Pre-emptively divert call?

Y

Pass call through to user

N

User able to answer call?

Y

User takes call

N

Pass call to voicemail

Record details in call database S1
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/60 H04M5/54

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED
Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F H04M

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic database consulted during the international search (name of database and, where practical, search terms used)
EPO-Internal, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>WO 98 38781 A (NORTHERN TELECOM LTD ;WILL CRAIG ALEXANDER (US)) 3 September 1998 (1998-09-03) the whole document</td>
<td>1-26</td>
</tr>
</tbody>
</table>

X Patent family members are listed in annex.

Further documents are listed in the continuation of box C.

* Special categories of cited documents:
* A* document defining the general state of the art which is not considered to be of particular relevance
* E* earlier document but published on or after the international filing date
* L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
* O* document referring to an oral disclosure, use, exhibition or other means
* P* document published prior to the international filing date but later than the priority date claimed

Date of the actual completion of the international search
7 October 2002

Date of mailing of the international search report
30/10/2002

Authorized officer
Sündermann, R

Form PCT/GB/510 (second sheet) (July 1992)
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AU 6337598 A</td>
<td>18–09–1998</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 0962088 A1</td>
<td>08–12–1999</td>
</tr>
</tbody>
</table>