Theoretical Foundations for User-Controlled Forgetting in Scrutable Long Term User Models

Debjanee Barua, Judy Kay, Bob Kummerfeld
School of Information Technologies
University of Sydney
Australia, 2006
{dbarua,judy.kay,bob.kummerfeld}@it.usyd.edu.au

ABSTRACT
Emerging technologies are making it feasible for people to capture large amounts of personal information that can support important aspects of their lives. When that information is stored in a persistent storage so that it can drive personalisation, it is called a user model. There has been little exploration of systematic approaches to enable people to gain control over such user models using forgetting mechanisms. This paper first presents reasons why such forgetting mechanisms are desirable. Then it analyses core forms of human forgetting as the basis for a theoretical model of forgetting in user models. Our key contribution is to establish theoretical foundations for the design of mechanisms and interfaces for forgetting in stores of personal information, with the goal of user control of these mechanisms, so that we can enable people to achieve a new form of control over their personal information and its use.

Author Keywords
Pervasive and ubiquitous computing, Personal information stores and its use, Long term user model, Scrutability and user control, Intelligent user interface, mechanisms of forgetting

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: Miscellaneous—Optional sub-category

INTRODUCTION
There is a fast growing usage of pervasive and ubiquitous computing with computational power embedded in everyday digital objects that encourages the collection and movement of huge amounts of data that are about us and belong to us. This computational power provides the potential to exploit these data to enable our intellectual development and learning, and to improve our physical health and social well-being, particularly when used as a lifelong resource. For example, people can now use technologies like FitBit\(^1\) that tracks daily physical activity of the user, Slifeweb\(^2\) that logs user activity on desktop and web, or lifelogging technology such as MyLifeBits\(^3\). Personally controlled health care systems (e.g., Microsoft’s HealthVault\(^3\)) are enabling users to hold and maintain their medical and health records. When the collection of personal information captured by these tools is stored in a persistent storage and used as the basis of personalisation, it is called a user model. Since the user model holds personal information about a user, that particular user should have control over it. One reason for this relates to control over the privacy of this information\([20]\). User control over personal data has received considerable attention in emerging pervasive computing frameworks\([3, 16, 24]\), especially in health domain\([6, 26, 27]\). One neglected, but important, aspect of control over personal information is that of forgetting. This has been noted as a concern that needs to be addressed, for example in the context of lifelogging\([13]\) and social networking\([12, 34]\).

In the next section, we describe the nature of the forgetting mechanisms that we aim to support. Then we summarise key ideas from research on human forgetting and show how we used these to inspire our thinking about a theoretical model of user-controlled forgetting in user models. We then compare this model against related work and conclude with a summary of our goals for user control over personalisation and the personal information in user models, and the ways that our model provides a foundation for them.

USER MODELS AND FORGETTING
User models are at the core of long term personalisation in ubiquitous computing environments since they determine a particular behaviour of personalised applications for a particular person. User modelling research is based on the view that a user model should be represented as a set of information about the user, separate from any particular application. If a pervasive computing environment is to personalise its services, its applications need to make use of relevant information about the user, from the user model.

Since user models hold personal information about the user,
these should be **scrutable**, meaning that the users can, when they wish, scrutinise or monitor the user model to determine what information it holds about them. This is a foundation for enabling the users to control their model and its use, and so to control the personalisation processes within pervasive computing.

Figure 1 illustrates our vision for the role of a user model in people’s long-term personalisation. At the top middle, we show the many and varied devices in personal digital ecosystems. At the left, we show that these accumulate huge amounts of personal information in various silos, on various machines, and on cloud services. We envisage that a person should be able to gain control of this information in their own scrutable user model (at the bottom left of the figure) using an effective and easy-to-use user interface (at the bottom middle of the figure). Like the existing stores, this would be used over long periods of time through a person’s life. The lower right of the figure shows how this scrutable long-term user model can serve several roles. First, it can be made available to applications, and enable them to personalize their interactions with (or on behalf of) the user. The second role for the user model, when presented in a suitable form, is as an aid for reflection. The third potential benefit is to support augmented cognition and life-long memories, by providing people with ready access to pertinent information about themselves as needed.

Consider the following scenario:

Alice is health conscious. She uses several devices to help her achieve health goals. One is a Fitbit which makes it easy to track her physical activity and sleep. She also uses an application that helps her identify posture problems, learn exercises from online tutorials and record the exercises she does. Her scrutable long-term user model aggregates data from sources such as the Fitbit and the application. This long-term user model has an interface that helps her monitor what information it holds. Alice can also input information directly into her user model using this interface. If her brother David borrows the Fitbit for a week, Alice can remove that week’s data from her user model. She herself sometimes inputs some invalid information mistakenly and later wants to remove it from her model. She wants to summarise or aggregate the daily data from the sensor or monthly exercise records from the application as it might help her reflect on her activity without showing fine-grained data. She visits the gym and shares information from the Fitbit and posture application with her personal trainer. She does not want to share all the data from Fitbit. She wants to hide some parts of the user model and selectively disclose the desired part to the trainer. Alice has preserved all her personal data in her user model for 5 or 6 years and finds that her system has become very slow. Now she has become expert in several exercises and does not need to go through tutorials about them. So she needs her system to remove that information gradually by setting a **decay rate** over so that it is still available but in a less readily accessible form. She also decides to archive some of the rarely used data from previous years to a back-up storage as she does not need to access it currently. She starts to use a new sensor for tracking her physical activities and occasionally uses her old Fitbit. Usually these two sensors unobtrusively store data in the user model. Now she wants to have the option to **uninstall/block** one sensor application, perhaps preventing it from storing data in her user model. Alternately, if data is accrued from that device, she wants to set some rules to **block** that data from being accessed by other applications. Sometimes she needs some automated support from her system to help with the above tasks, such as aggregation of data every month or decay of very old data which has been unused over a long period of time.

The above scenario illustrates why we might need forgetting in scrutable long-term user models. In fact, it is important at several levels. At a purely technical level, as we can see in this scenario, after using the system for a long period of time, large information stores can slow down it. Even more importantly, in cases where Alice needs to explore and understand a part of her user model, it may be more difficult if it is large and complex, with outdated or unwanted information to sift through. She needs to have some mechanism to reshape or reform information to a different granularity level. Even more fundamental is the issue of ensuring that people are able to have practical control over such information; so if Alice decides that parts of the information should not be kept or that it should only be kept on certain devices, she should be able to effect this form of forgetting. This may be for the very simple reason that the stored information is incorrect. So the central notion of implementing forgetting in long-term user models is that a person should have control over their own user models, including the right to have parts removed.

In this paper, we present a theoretical model to inform the interface design for a software framework with mechanisms enabling people to control the forgetting of their information from their user models. The framework builds upon the Personis framework [3] which was designed, with the explicit goal of ensuring that its representation should enable users to scrutinise and control their own user models in ubiquitous computing environments over long periods of time. Its design has support for a person to determine what informa-
tion is allowed into their model and what applications may use which parts of it. This paper explores possible forms of forgetting implemented in such a long term user modelling framework, what the user should be able to do with these mechanisms and how they may achieve this. In [7], we presented the theoretical model underlying the technical implementation of forgetting mechanisms based on the accretion/resolution representation of Personis framework. This paper moves to the interface issues.

The next section introduces our theoretical model which analyses potential benefits and user interface implications of different forms of forgetting from HCI perspectives.

THEORETICAL MODEL

Our theoretical model has two parts. The first is based on the literature on different mechanisms of human forgetting. This helped us determine the benefits of these mechanisms in the scrutatable long term user model. It also enabled us to identify the interface implications of potential mechanisms for user-controlled forgetting. The other part of our theoretical model comes from an HCI perspective that includes different user views for enabling a user to exercise the mechanisms, the consequence of implementing each of these mechanisms and what support the users will need if they make a mistake which we call the “Back-up plan”.

Forgetting in human memory

Since our work endeavours to establish mechanisms similar to the human phenomenon of “forgetting”, in this section we explore psychological theories that focus on the contribution of forgetting in human memory as an inspiration to our work. Apart from identifying the forms of forgetting, our analysis in human forgetting helps us find out the factors that affect forgetting in human memory and design a user interface that enable users to control these factors and consequently control forgetting in their long term user modelling system.

In human memory, forgetting is usually viewed as a regrettable loss of information. But some psychologists have suggested that forgetting may be functional. One of the first to explore this possibility was James [17], who wrote, “In the practical use of our intellect, forgetting is as important a function as recollecting.” According to his view, forgetting is the mental mechanism behind the selectivity of information processing, which in turn is “the very keel on which our mental ship is built.” Forgetting helps human memory avoid information overload and update information as necessary. It prevents retrieval of information that is likely to be obsolete and thus helps people focus on current goals [1]. Baddeley [4] suggests that “The process of forgetting is one whereby the important features are filtered out and preserved, while irrelevant or predictable detail is either destroyed, or sorted in such a way that it is not readily accessible in its original form.” These indicate the benefits of forgetting in human memory and provide a source of inspiration for the design of mechanisms for forgetting in long term user models.

One dimension of research in human forgetting focuses on spontaneous [8] or incidental [5] forgetting in human memory which occurs without motivation and regardless of the information’s validity or relevance. This form of forgetting implies that forgetting in long term models might be an automated process. Table 1, summarises the forms of spontaneous forgetting. Table 2, in contrast, presents another dimension of human forgetting called motivated forgetting. This includes intentional forgetting processes initiated by a conscious goal to forget and directed forgetting that needs instruction to forget [5]. This category suggests the class of forgetting which might be controlled manually by the user. Referring to our scenario, Alice might have several options in the user interface for her user modelling system. One is an automated process that operates unobtrusively once activated by her. Another is a manual process where every time she implements the mechanisms manually. Some mechanisms of human forgetting have been considered in both incidental and intentional forgetting. These are explored in Table 3. These three tables also include potential implementation and benefits of these forms of forgetting in long term user models and their possible implications for user interface.

Baddeley discusses the most important factors that cause forgetting to occur in human memory in [5]. This helps our analysis of human forgetting to come up with the concept of drivers (Column 1 in Table 1, 2 and 3) where this shows the primary factors related to each of these forms. One key role for such drivers is to help us determine the conditions when the user might want to forget and also determine the user interface supports the user might use to regulate forgetting in user models.

Time is an important driver of forgetting as memory traces may become weaker over time (G1-Table 1) or the number of interfering events increases causing a cue-overload problem over time (G2-Table 1). The first group of forgetting mechanisms (G1-Table 1) might occur due to trace decay or transience or fading over time, as is often observed as time elapses between learning and remembering [8, 33]. This also deals with the strength of information, as less rehearsed items become weaker and more prone to this phenomenon. Decay is an important mechanism for forgetting in user models, as it offers ways to gradually remove obsolete information, thus keeping the user model up-to-date. Referring to our scenario, Alice wants to gradually reduce the accessibility of previous exercise records. So this form of human forgetting suggests that we should consider time and the strength of information for gradual forgetting with a particular decay rate. And the user should have the option to control the decay rate and the activation of automated gradual forgetting.

Another important aspect is interference which causes retrieval of memory to be disrupted by related traces in memory. Increasing interference between earlier and later learned information in memory can cause forgetting. Newly learned items are blocked by older items in proactive interference and older information is blocked by new information in retroactive interference [5] (G2-Table 1). This is also an important mechanism for forgetting in user models as the user should
<table>
<thead>
<tr>
<th>Driver</th>
<th>Group ID</th>
<th>Forms of Human Forgetting</th>
<th>Forgetting Process</th>
<th>Potential implementation, benefits of forgetting and possible interface implications in long term user models</th>
</tr>
</thead>
</table>
| Time and strength              | G1       | • Fading  
• Transience  
• Trace decay                                                      | Decay process: Automatic fading of memory traces and gradual forgetting over time [8]. | • Implementation: Mechanisms for gradual removal of old and obsolete information based on use of the information.  
• Benefit: Important. Reduces storage requirement of active user model and increases speed of access.  
• Interface: User control over decay rate and activation of automated forgetting. |
| Time and Interference          | G2       | Retroactive interference or Proactive interference                | Blocking and unlearning: Forgetting due to dominating interference of new or old information can block access to or unlearn older or newly learned information respectively [5]. | • Implementation: Mechanisms for detecting and blocking interfering obsolete data.  
• Benefit: Important. Increases the speed of operation by blocking unwanted parts.  
• Interface: User control over power of source, importance and usage of interfering information. |
| Interference                   | G3       | Retrieval induced forgetting based on “strength dependent competition model” of interference | Retrieval process: Repeated retrieval of a given item will strengthen that item while reducing the recall probability of other competitor items [2, 5]. |  |
| Time, Interference or Catastrophe | G4       | Trace dependent forgetting                                        | Destruction or reformation due to decay or unlearning process: Retrieval failure as stored information is not available at all or in its original form due to deterioration/reformation by subsequent/previous learning [5, 25]. | • Implementation: Mechanisms for removal of obsolete data, misconceptions and errors is possible.  
• Benefit: Important. Keeps the user model less prone to errors.  
• Interface: User control over identifying data to be deleted and summarisable. |
| Lack of proper cues            | G5       | Cue dependent forgetting                                           | Retroactive inhibition: Retrieval failure due to unavailability of inhibited cues [35]. | • Implementation: Activation or deactivation process of different resolvers/rules for interpreting the data depending on the context.  
• Benefit: Important. Provides flexible ways for data interpretation and helps avoid unwanted data given the context.  
• Interface: User control over conditions stimulating to-be-forgotten data. |

Table 1. Spontaneous or Incidental forms of human forgetting, implementation, benefits in long term user models and interface implications
### Table 2. Motivated or Intentional or Directed forms of human forgetting, implementation, benefits in long term user models and interface implications

<table>
<thead>
<tr>
<th>Driver</th>
<th>Group ID</th>
<th>Forms of Human forgetting</th>
<th>Forgetting process</th>
<th>Potential implementation, benefits of forgetting and possible interface implications in long term user models</th>
</tr>
</thead>
</table>
| Importance, Secrecy or emotional attachment | G6       | Goal directed or emotional forgetting | Suppression: Blocking of some stored information intentionally; Repression: Unconscious defensive process to exclusion of unwanted memories. [5] | • Implementation: Mechanisms for blocking of information inhibits pertinent access to the otherwise unimportant information.  
  • Benefit: Important. Preserves privacy.  
  • Interface: User control over visualisation and process of identifying what to block. |
| Instruction to forget               | G7       | Forgetting during learning | Set differentiation: Functional segregation of to-be-forgotten item from to-be-remembered items [31]. | • Implementation: Mechanisms for setting and updating some parameters on the personal data during remembering and recalling processes.  
  • Benefit: Relevant. Provides scope for prioritisation of information in the user model in the given context.  
  • Interface: User control over setting importance and track accesses of the context and components while storing in the model. |
|                                     |          | Forgetting during rehearsal | Selective rehearsal: To-be-forgotten items are not or less rehearsed [31]. | }

### Table 3. Both incidental and motivated forms of human forgetting, implementation, benefits in long term user models and interface implications

<table>
<thead>
<tr>
<th>Driver</th>
<th>Group ID</th>
<th>Forms of Human forgetting</th>
<th>Brief Description</th>
<th>Potential implementation, benefits of forgetting and possible interface implications in long term user models</th>
</tr>
</thead>
</table>
| Retrieval inhibition               | G8       | Retrieval induced forgetting due to “suppression or inhibition” of nontarget items | Retrieval of target items requires inhibition of nontarget competitors which eventually causes impairment of those inhibited items [2, 35]. | • Implementation: Mechanisms for inhibition of interfering information during access.  
  • Benefit: Important. Blocks unimportant information.  
  • Interface: User control for identifying and blocking interfering data. |
| Time and Context change            | G9       | Context change account of forgetting | Contextual fluctuation: Forgetting due to the mismatch between learning context and retrieval context blocks information [8, 31]. | • Implementation: Mechanisms considering context while implementing forgetting.  
  • Benefit: Important. Preserves privacy and accounts for value of information.  
  • Interface: User control over visualisation and manipulation of context. |
be able control ways to forget interfering information. For example, in our scenario, if Alice’s old Fitbit is activated, it will store worthless physical activity records (e.g. Steps = 0, Activity=Sedentary) in the user model. If an application retrieves only the last 20 physical activity records, this will include the records from the older Fitbit instead of some important values from the new sensor. This is indeed a form of interference with the data from her new activity sensor. To avoid this, Alice should be able to specify the required data source for the records and thus block the interfering unwanted information. So this class of forgetting suggests that users should understand and consider the identity and power of the source of the information and the importance of a particular piece of information while implementing forgetting.

**Interference** during the retrieval process is the driving force for “Retrieval induced forgetting” (G3-Table 3). Here, strong competitors are retrieved repetitively, reducing the retrievability of weaker target traces [2, 5]. This role of human forgetting is important for long term user models as this suggests the superiority of frequently accessed data over less accessed data. For example, Alice decides to model the activity of some of her colleagues and her posture application accesses that regularly for comparing her progress with others. After some time, two of her colleagues leave and stop sharing their data with her system. So the system gradually strengthens the accesses for the existing models while make the system forget the models of those two colleagues as these cease to be accessed. Thus the user interface should provide a visualisation and explanation of useful information from the user model.

Insufficient and inappropriate retrieval cues cause **Cue dependent forgetting** (G5-Table 1). This means retrieval fails due to the inaccessibility of information. The inaccessibility of cues may occur for retroactive inhibition [35], while irrelevant cues hamper retrieval due to the change in contextual cues over time [8]. This form of forgetting is relevant for user models as it suggests that a user might be able to activate or deactivate the tools that retrieve and manipulate the data in a given context. To achieve this, the user needs support to understand the task and process involved and consequences of activating and deactivating them.

The importance of information drives intentional forgetting due to **suppression** of goal directed exclusions of memory from awareness (G6-Table 2) [5]. Secrecy of unwanted memory can be preserved by repression where unpleasant events are blocked by defensive process of human mind. This class of forgetting suggests that users should have ways to block irrelevant and private personal data in the user model. Another form in this category is the instruction to forget information. This (G7-Table 2) can cause directed forgetting by set differentiation and selective rehearsal [31]. This reflects the benefits of prioritisation of information during learning and remembering processes in the user model which results in intentional forgetting of unwanted information. As in our example, Alice sets a goal of loosing weight. She might be able to create a new data-set with associated subgoals (e.g. walking, eating vegetables etc.). The information that is outside the set is still in the user model, but will not be accessed. It will remain forgotten for some contexts. Also, with such mechanisms, Alice can hide particular information from her trainer. So the user interface should provide options to enable the user to create goals, link related data-sources to these goals and block a particular data-set as desired.

Other research on both incidental and directed forgetting suggests that a key mechanism of forgetting, as an updating process, is retrieval inhibition (G8-Table 3) [2]. Taking this view, an inhibitory mechanism is invoked at the time of retrieval. This reduces access to unwanted memories, producing lower recall of those. Because these memories are less accessible, they are correspondingly less likely to interfere with recent memories, producing the benefits of directed forgetting. An alternative account suggests that both spontaneous [8] and directed [31] forgetting may be due to context change effects (G9-Table 3) with the passage of time. Context dependent forgetting is very important as it also blocks information that is irrelevant in the current context, temporarily forgetting that part of user model. This also relates to the scenario where Alice is trying to achieve a new weight loss goal.

Thus we can see that theories of human forgetting help us to identify possible roles of forgetting in long term user modeling and the factors to which users need access to control forgetting. Most importantly our research focuses on decay, deletion, compaction, inhibition, and finally archival of data from the user models. The next section introduces what these mechanisms will offer, how users will be able to use them and what the outcomes will be of each operation including some back-up plans for recovery.

**User interface for forgetting in long term user model**

Since this paper focuses on designing interfaces for controlling the new forgetting mechanisms, we must consider ways to make this more user friendly so that users do not become overwhelmed with new functionalities and are able to exercise the mechanisms effectively with their current technical knowledge. Our interface will need to provide necessary digital artefacts to implement each of the five possible mechanisms namely decay, deletion, compaction, blocking, and archival that we identified in last section taking inspiration from human forgetting analysis. We have developed a theoretical model of the storage hierarchy from highly ac-
<table>
<thead>
<tr>
<th>ID</th>
<th>Short title</th>
<th>Role for forgetting in long term user models</th>
<th>Possible user interaction</th>
<th>Possible consequences of action</th>
<th>Possible backup plan for recovery from accidental action</th>
</tr>
</thead>
</table>
| 1  | Decay       | Gradual decrease in accessibility of information | - Visualisation and control over the decay rate of a particular piece of information;  
- Initiation of automated gradual forgetting. | Less used or accessed. | Options for changing the decay rate at any point. |
| 2  | Delete      | Deletion of erroneous evidence or misconceptions | - Visualisation of erroneous or obsolete information;  
- Deletion of unwanted information. | Data is lost forever; System is unable to answer historic queries. | Undo Deletion; Stored in remote server with a default decay rate before complete deletion and can be restored. |
| 3  | Compaction  | Summarizing part of model and replacing that part with a more compact set of information | - Visualisation of summarisable pieces of evidence;  
- Summarisation of those pieces of evidence in one single evidence followed by deletion;  
- Setting a general rule for the compacted evidence. | Loss of fine grained detail. | Undo Compaction; Detailed part is archived in remote server with a default decay rate before complete removal of detailed part and can be restored. |
| 4  | Blocking    | Blocking access and disclosure of information | - Visualisation of data or processes or identity interfering sources of information to be blocked;  
- Blocking of unwanted/secret components or evidence from being accessed by particular application or device;  
- Blocking of interfering sensors. | Not accessible by other applications and devices. | Possible to disable the rule/system for blocking or set new rules any time. |
| 5  | Archival    | Archival of parts of user model to a backup store that have no or less contribution to the reasoning process | - Acquiring a back up storage in a remote server;  
- Moving less important context/components/evidence to the back-up store | Not actively take part in the reasoning process. | Can be restored if an application needs the archived information. |

Table 4. Forms of forgetting in long term user models, user interaction, consequences and back-up plans of different forms of forgetting
cessible to less accessible storage devices. These are an active model for holding useful and current parts of the model, an archived model holding the less used parts and finally a deleted model which holds the deleted and obsolete parts of the model. Each of the active and archived models holds a separate region for blocked information. We will refer to these storage models in the rest of this section.

Table 4 gives an overview of these possible forms of forgetting in long term user model from the user perspective. It describes what the user will be able to do and how they will do it. To implement these mechanisms effectively, the user should understand the consequences of each action and the recovery plan for each of these mechanisms. Hence Table 4 also addresses these issues.

**Decay:** Decay is considered to be an important form of forgetting which occurs with the passage of time (G1-Table 1) considering the strength of the information (G3-Table 1). In our theoretical model, it means that the old and unimportant part of the model will be moved from one part of the model to another through the storage hierarchy and eventually, deleted over time (Row 1 Table 4). With this mechanism, Alice can remove some parts of her model (e.g. the activity models of two colleagues) from her active model. For this she needs to manually control the decay rate of the information or initiate an automated process that will gradually remove those from her user model. The removed information will be less accessed by the external application and internal processes. The user interface will provide four options for setting a decay rate (i.e. none, slow, moderate and fast) to the information in question. Considering our scenario, Alice can set or change the current rate to any of these four at any level of the storage hierarchy. Before transferring information from one level to another, she will receive a message/prompt from the system about the action.

**Deletion:** Complete deletion of unwanted data is a useful mechanism for forgetting in user models (Row 2 Table 4) which is quite similar to the mechanism for destruction (G4-Table 1) or unlearning (G2-Table 1) of traces in human memory considering the strength of information (G3-Table 1). With this operation available, Alice can delete the irrelevant data gathered from the FitBit or wrong data entered by mistake. But the consequence of this operation is highly critical. Because once the information is deleted, she will not be able make historic queries on that. This might have a negative effect if she accidentally deletes something important. So before deletion she needs to consider which other applications should be affected by this deletion operation. The deleted information is kept on a remote server in the deleted model with a default slow decay rate. Before the final deletion, she will receive a message from the system asking whether she wants to restore the information or use any other forgetting mechanism so that the information is not lost forever.

**Compaction:** Compaction means summarising detailed information into a more compact set of information (Row 3 Table 4) and then transfer the detailed part down to the slower storage hierarchy. This can also be related to reformation (G4-Table 1) or unlearning (G2-Table 1) of weaker memory traces in human memory over time. In our scenario, Alice can use this mechanism to aggregate her daily exercise records into one monthly record. To accomplish this, she will use interface supports to find out which information can be summarised and apply this tool for summarise the data into one single information. Since aggregated information is not available in its original form, it might not be possible to answer some historic queries in fine grained detail. Rather the summarised evidence and associated rule can be used for making any general inference about the user in that context. For example, her summarised exercise records might answer her overall performance and improvement queries on a monthly basis. This can also have a similar negative consequence as deletion. So as a back-up plan, the detailed, removed parts of the compacted information will be kept in remote server in the archived model with a default decay rate to transfer it to the deleted model and will be completely lost afterwards. At each level of transfer and before final deletion, Alice will be prompted to ask if she wants to restore the information in its original form.

**Blocking:** Blocking is a common mechanism in human forgetting [33]. It encourages blocking of unwanted private (G6-Table 2), interfering (G3-Table 1) or irrelevant information for current context (G5-Table 1, G9-Table 3). In case of long term user models, it means blocking the disclosure of some parts of model which are not relevant but would compete for the user attention with the information in the current context (Row 4 Table 4). This also supports hiding some private parts of the user model from external applications and devices. Blocked information is kept in the blocked region of the active model and might be decayed to the archived model over the long term. This will not be used for answering historic queries. In our scenario, Alice will be able to block disclosure of some parts of her model while sharing with her trainer or block access to the data while using one of the two activity sensors. To exercise this mechanism, the user interface should allow Alice to see the information that she needs to block and activate some rules/systems for this operation if necessary. She can disable the rules/systems and make the information accessible for external and internal processes.

**Archival:** As in human memory, some information in the user model becomes irrelevant over time due to contextual changes (G5-Table 1, G9-Table 3). Such unused or old information information could be archived in the archived model (Row 5 Table 4). As a consequence, archived parts of user model are not accessed by other applications. But unlike blocked parts, they can be used by any application in the given context. For example, if Alice archives the part of her user model that contains the exercise records of years ago, these become inactive and are not readily accessible by the external application. If an application requires access to that part, it might retrieve it from slower storage with necessary authentication. If the archived part is required frequently by an application, then Alice can move it back to active user model at any point.
In the next section, we outline the previous work in forgetting in computing systems and relate it to our present theoretical model.

RELATED WORK
Research in forgetting in computing systems is a new area, with much yet to be explored. Here we review some of the work that has been conducted, especially from an HCI perspective. As we discuss this literature, we refer to rows in the table of our conceptual model in Table 4.

Substantial work has been done in machine learning where forgetting was used to deal with “Concept Drift” in machine learning [23, 28]. This is highly relevant to our work as concept drift is an acknowledged problem in user modelling, as people change. Particularly salient is the work on “Gradual forgetting” [23] (Row 1 Table 4; G1 in Table 1), where the algorithm learns to identify changes in concepts so that irrelevant examples are deleted gradually and the system is trained again with the remaining examples. This research is motivated by purely technical concerns and has not developed user interfaces nor performed usability studies. But it supports our motivation and theoretical model for forgetting.

Forgetting has never been considered in long term user modelling, particularly from an HCI perspective. User modelling systems such as UMT [9], LDAP based user modelling system [21], BGP-MS [22], and TAGUS [30] offer some graphical tools that allow system builders to build, edit and manipulate the details of user models. These systems offer some support to create, update and delete user models to which end-users have no direct access. For example, in UMT [9], tools like a consistency manager resolve conflicts by removing (Row 2 Table 4) or hiding (Row 3 Table 4) the parts of the user models that include the contradictory assumptions. TAGUS [30] implements core maintenance functions to provide a set of functionalities, e.g. contraction for deleting (Row 2 Table 4) information from the user model. The LDAP based user model [21] offers support for back-up (Row 5 Table 4) for archiving the web-interactions of the user. This system also supports deletion (Row 2 Table 4) following standard mechanisms of LDAP protocol and blocking (Row 4 Table 4) in the form of base-filters. Since these user modelling systems are not scrutable, such facilities are not available to their existing user interfaces for direct user control. The interface of the Doppelgänger user modelling server [29] provided visualisation tools for end users; but this did not allow users to control pruning unimportant or aging information (Row 2 Table 4), or reducing the reliability of sensors (Row 1 Table 4).

One of the key roles for Open Learner Models (OLM) is to enable the user/learner to contribute to the model to improve the accuracy of the model. Different interfaces for OLMs have been explored for large user or learner models. Some of these provide sophisticated graphical user interfaces for learner control for deleting (Row 2 Table 4) misconceptions or errors [10, 15, 36] to resolve inconsistency and blocking (Row 3 Table 4) information to share with peers and tutors [11, 18, 36] to maintain privacy. These have also been evaluated for effective user control. But none of them explicitly evaluated the user interfaces for supporting the forgetting mechanisms that we proposed in the last section.

Our work is based on the Personis user modelling framework that was designed to support user control of user modelling [3]. This system supports scrutiny control management [19]. It enables a user to define new parts of the model, applications that may use it, and the access mechanisms for each of these applications. Its context-dependent filters are designed for privacy, controlling which sources of evidence are allowed for a particular application and thus supports Blocking (Row 3 Table 4). This interface might be augmented with the mechanisms for forgetting discussed in last section in Table 4. For example, if a user wants to effect decay (Row 1 Table 4), he/she can exploit the use-by date mechanism.

DISCUSSION AND CONCLUSIONS
The goals of this research are to define theoretical foundations from an HCI perspective for the design of a user interface to support forgetting mechanisms and implement these mechanisms in our framework for long term user modelling. Our model aims to capture two key elements to drive our future work:

- The forms of human forgetting that can have potential benefits in long term user models and their implications for using in a long term user model interface in pervasive and conventional computing applications.
- The forms of forgetting that should be supported by the user modelling systems, how the user can potentially manipulate these forms, the consequences of each operation, and the mechanisms for recovery in case of an accidental forgetting operation.

A key challenge for giving people control over their user models is to design a user interface to support these mechanisms. This paper serves as a foundation for that enterprise by identifying the nature of the forgetting mechanisms to create and for which we need to create interfaces.

REFERENCES


