This chapter explores the potential for improving long term learning by exploiting the large amounts of data that we can readily collect about learners. We present examples of interfaces to long term learner models that illustrate both the challenges and potential for them. The challenges include the creation of a suitable technical framework as well as associated interfaces to support a learner in transforming arbitrary collections of learning data into a lifelong learner model. We explain how this can provide new ways to support both the learner and applications in personalizing learning over the long term, taking account of the individual’s long term development.

Lifelong learning deals with the full breadth of learning, be it in a formal classroom or outside (Longworth, 2003), over long periods of time. This contrasts with the current norm for educational technologies, where the learner interacts with many computer-based tools, each operating independently of the others. Each collects its own data about the learner; this means that there are many independent silos of information about the learner and their learning progress. Commonly, one learning tool holds the data associated with learning a very specific skill, or a single subject, perhaps running over weeks or months. Such silos reduce the possibility for making use of this aggregated data over the long term.

A particularly important aspect of lifelong learning relates to the need to support self-directed learning (Candy, 1991), and learner autonomy (Goodyear, 2000), especially in the case of adult learners. There appears to be a potentially important role for wayfinding systems which can help the learner plan their learning path at each key stage (Tattersall et al., 2005). Personalized learning also has the potential to be important for lifelong learning (Knapper and Cropley, 2000) as does support for collaborative adaptive learning (Klamma et al., 2007), with learner control and choice (Janssen et al., 2007).

This chapter explores the ways that a lifelong learner model can play this
role, where we define it as a store for the collection of learning data about an individual learner. To be useful, it must be able to hold many forms of learning data, from diverse sources. It must be able to make that information available in a suitable form to support learning.

Consider the following scenario:

Alice, a 40 year old pediatrician, has many interests, including playing social netball, primarily to help her keep fit. At one netball training session, the coach recommends NetCoach, a new online training system for netball players. It also provides a smart-phone application to help monitor daily exercise and performance.

This scenario illustrates several forms of long term learner modeling. First consider Alice’s long term goal for fitness. This is, indeed, a lifelong goal with a complex of elements and it is typical of some of the most important of people’s goals, remaining relevant for much of their lives. Alice’s long term model might capture considerable information that is useful for helping her achieve that goal. For example, it could model her knowledge about fitness, about exercises, how to do them correctly, effective training regimes, ways to overcome injuries and about healthy eating. Success in achieving these is reflected in various measures of health such as weight, blood pressure, strength, cardiac fitness and performance on physical activities. A system like the hypothetical NetCoach might reuse an existing part of Alice’s lifelong learner model. For example, there may be long term models of her knowledge about fitness and exercise as well as performance data reflecting her changing fitness. This could drive personalization of the coaching. It might also reuse part of her model for past sporting activity and interests.

While a program like NetCoach might reuse parts of her lifelong model, another important role for the model is to support Alice in reflection on her long term progress and as an aid for her planning new goals. She may also want to be able to share parts of these models with other people, such as her netball coach. In a sense, this would enable the human coach to reuse part of her long term model. For example, if the coach can see the impact of new learning materials and new training programs on her long term fitness, this could help the human coach to help Alice devise her plans for improved training and learning. If the coach can aggregate learner models for all the members of all the teams they coach, they can analyze this to improve their understanding of their coaching methods.

A second class of long term learner model is associated with her knowledge and skills in medicine and pediatrics, from her university days, through the range of her postgraduate specialist education and ongoing training activities and work experience. Alice needs to maintain and develop her expertise
as a pediatrician, at the highest level she can. This is representative of an important class of long term learning: that required to achieve expertise. It demands long term *deliberate practice*, “a highly structured activity, the explicit goal of which is to improve performance. Specific tasks are invented to overcome weaknesses, and performance is carefully monitored to provide cues for ways to improve it further” (Ericsson et al., 1993). A key characteristic of deliberate practice is its long term nature. This is because it requires “effortful activity that can be sustained only for a limited time each day”. So, it takes many days, of concerted effort, to reach high levels of achievement. Ericsson observed that it requires activity that it is “not inherently motivating”. Rather, the learner does it to achieve “improvements in performance”. High levels of achievement also appear to require excellent teaching or coaching (Ericsson et al., 2007). This means that Alice should be able to use a diverse set of learning tools. Her lifelong learner model should *aggregate* model information from these. Then the lifelong learner model can allow many different learning tools to *reuse* parts of the model. At the same time, if there are suitable interfaces to the right parts of the learner model, they can help Alice *reflect* on her long term learning and *plan*.

The next section establishes a foundation for the new challenges for lifelong learner modeling. It identifies new roles that the lifelong learner model can fulfill and, for each of these, it explores the technical challenges still be be addressed. The following two sections present selected examples of our explorations of two of the most important of these. We conclude with a discussion of the particular challenges for lifelong learner modeling.

### 1.1 Roles for lifelong learner models and technical challenges

The motivation for lifelong learner modeling is to address important learning needs that are currently unmet. We now identify and discuss these new key roles for the lifelong model. They are summarized in the left column of Table 1.1. In the scenario of the last section, we introduced examples of these. The right column lists the key new technical challenges associated with each goal. Some technical issues are important for multiple roles: *User interfaces* appears against each role, though the associated demands on them differ; *middleware infrastructure* also appears for most roles, but with different aspects required for different roles; *ontologies and standards* are important for all sharing and reuse as well as some forms of aggregation.

Before we discuss each of the roles, we present an architectural overview of the middleware infrastructure and its relationship to many applications that might play a role in supporting a user’s long term learning. Figure 1.1 shows
Table 1.1 *Roles for the lifelong learner model and technical challenges.*

<table>
<thead>
<tr>
<th>Role of learner model</th>
<th>Technical challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Open learning modeling <em>(OLM)</em></td>
<td>user interfaces for reflection and planning, attention, forgetting</td>
</tr>
<tr>
<td>2. Aggregation of information about the learner from diverse sources</td>
<td>middleware infrastructure for aggregation, user interfaces to control aggregation, ontologies</td>
</tr>
<tr>
<td>3. Sharing the learner model with people</td>
<td>user interfaces, particularly for privacy management, middleware infrastructure for control of privacy and security, ontologies</td>
</tr>
<tr>
<td>4. Interpretation of learner information</td>
<td>user interfaces, new tools for different interpretations</td>
</tr>
<tr>
<td>5. Reuse by different applications for the learner’s personal use</td>
<td>user interfaces, middleware infrastructure for controlling release of parts of the model and active delivery of parts; ontologies, standards</td>
</tr>
<tr>
<td>6. Institutional use of long term learning data from many learners</td>
<td>user interfaces, middleware infrastructure associated with both sharing and reuse of the model, ontologies, standards</td>
</tr>
</tbody>
</table>

... a user at the center top, interacting with examples of several important classes of applications, which we discuss below.

The figure shows the lifelong learner model, and its associated middleware at the bottom. Although the figure shows the model as a single large block, the infrastructure should support distributed models; for example, a user may prefer that particularly sensitive information be keep only on their home computer. This might require the infrastructure to support disconnected operation, gracefully dealing with the situation where a computer is temporarily unavailable, for example when the user’s home machine has been turned off. Our Personis (Assad et al., 2007) provides this support for modeling.

But the lifelong learner model requires much more than this. Consider first the case where the user interacts with an application that provides personalized teaching, such as the Anatomy Tutor, which Alice from our scenario may have used for several semesters of her university studies. This is shown in the center of the figure, with two way interaction between the user and the *Anatomy Tutor.* There is also two way interaction with model, as the
Anatomy Tutor stores its information about the learner in the model, based on recent interactions and retrieves information about the learner, where some of this information might have been contributed by other applications. Note that the figure shows an app a small glue code application that manages the actual interaction between the lifelong learner model and the Anatomy Tutor.

Next along to the right, the figure shows a more conventional tool, such as a Learner Management System (LMS). In this case, we show that the information from the LMS might be harvested into the lifelong learner model. This would be managed by an app that makes use of a standard interface (API) to the LMS. Since current conventional tools cannot yet make use of the lifelong learner model, we show only one-way transfer of information into the model.

Next to the right, the figure shows a class of tool that the user interacts with over long periods of time, like the NetCoach of our scenario. This has two way interaction with the lifelong learner model. It also exploits an emerging class of learner modeling information; this comes from real time sensors which automatically capture information about the user. For example, there
are already many inexpensive devices that can unobtrusively capture data about exercise, including GPS-based tracking of running and heart rate measures (for example, Garmin watches http://www.garmin.com/), and weight (for example, www.withings.com/). These can send their data wirelessly to a web-based service. In this case, the app must interact with that service to harvest the learner’s data and aggregate it into the lifelong learner model. In the figure, we show a dotted line from the learner to the sensors since the user does not explicitly interact with it.

In all the cases discussed to this point, the learner has no direct interaction with the learner model. These are important, representing existing as well as emerging learning technology. We now consider three important classes of applications which require the learner to interact directly with the lifelong learner model. These are shown at the upper left quadrant of the figure. We begin with the one with a direct learning role, because it provides an interface which enables the learner to view their model. This is coming to be called an Open Learner Model (OLM). We discuss this important class below.

To this point, we have glossed over an important issue: the control over the apps that can interact with the lifelong learner model. Given the personal nature of the information within the lifelong learner model, we consider it essential to give the learner effective ways to control this and other aspects of the model. The management interface is a critical part of the lifelong learner model since it enables the learner to control which apps may interact with the model, so controlling which classes of information can be aggregated into the learner model and which applications may use and reuse the model.

The final class of tool shown is direct user input which enables the learner to add information directly to the model. For example, in our scenario, Alice might use this to inform the model of her fitness goals; such as her goal weight, running speed, rate of training.

Support reflection on progress and planning to achieve long term goals
As a learner strives to achieve long term learning goals, the lifelong learner model could play an important role in helping them monitor their progress and actively plan their future learning. This has been the focus of Open Learner Modeling (OLM) research (Bull and Kay, 2007). Table 1.1 shows three associated technical challenges.

First is the need for effective interfaces such as those from research in Learner Modeling for Reflection (LeMoRe 1). This has explored many approaches to presenting learner models and supporting interaction with them.

1 urlhttp://www.eee.bham.ac.uk/bull/lemore/
For example, Bull et al. (2007b) conducted eye-tracking studies with six different views of learner models. Students were reported to make extensive use of their open learner models, particularly exploring their misconceptions (Bull and Mabbott, 2006; Bull et al., 2008). Careful evaluations of OLMs have demonstrated important benefits. For example, in the SQL-Tutor, the open learner model led to learning gains for weaker students and was generally appreciated by the class as a whole (Mitrovic and Martin, 2007).

There has been little exploration of how to help the learner see what deserves their attention in the long term. In many cases, the most recent modeling information may be the most useful. However, other parts may be valuable. For example, suppose that a learner studied a single example repeatedly. An episodic learner model (Weber and Brusilovsky, 2001) which represents the importance of this example for the learner could ensure it is used as appropriate, even years later, by different teaching systems. Perhaps heavy use of a learning element, in a particular context, might make it stickier, a focus point in the long term learner model for that context. Another approach, inspired by human episodic memory (Craik and Tulving, 2004), suggests the value of a learner model representation for semantic connections to a particular concept in the model: rich connections may indicate importance within a particular context. The user’s own explicit input seems important, especially if this can be elegantly integrated into the learning environment. For example, a suitable interface might make it easy for the learner to mark examples they find most useful, perhaps tagging them with concepts that it helps to illustrate or recall.

Now consider the third issue, forgetting, both by the learner and the model. For the learner, we must account for cases where the modeling information is old, for example where there is an accurate model of the user’s mathematics knowledge at high school and they are now 50. One way to address this is to create mechanisms for the OLM to interpret old parts of the model appropriately. We will return to this below.

From the system’s perspective, should it forget very old modeling information, dating back many years? With low cost storage, it is easy to simply keep all the information in case it one day turns out to be useful. But bloated models with information that is never used may cause problems, particularly for privacy. It may also pose serious problems for scalability. This applies at the system level, as reasoning based on the model may become slow as the learner model grows. It may also become harder to create effective interfaces, since these may need to make the model understandable even as it grows larger. So the issue of forgetting may be important for creating
effective OLM interfaces. Others have made similar arguments for the value of forgetting for humans (Bannon, 2006; Dodge and Kitchin, 2007).

One intriguing but promising approach is to use a model of cognitive development, perhaps from theories of developmental psychology. More mundanely, but perhaps pragmatically, it might base a useful model on formal syllabus materials (which we might expect to have foundations in best practice, based on educational psychological theory). We have been exploring the latter, creating a pragmatics ontology for generic skills and their different levels then using this to describe university subjects, so that we can then create *curriculum maps* for complex university degree programs (Gluga et al., 2010). Our ontology design particularly aimed to support those responsible for the overall curricula to track how well each degree meets institutional goals as well as accreditation requirements. This required an ontology that represents development levels for generic skills. The subsequent models serve as a foundation for an individual learner model (Gluga, 2010) since they define a structure for the model, within the context of the university degree. To populate the individual model, we propose to draw upon the learner’s results in class assessments as well as other direct input from the learner and formative feedback from teachers. One can envisage similar ontologies and models associated with school curricula. One important benefit of this approach, which links the learner model ontology to developmental models, is the foundation it gives for interpreting older parts of the model. For example, it may be feasible to make interpretations such as *she was an excellent reader for a first grader*.

**Aggregation point for the information for the learner**

This is one of the key new roles that the lifelong learner model takes, as a long term repository of information about the learner, drawn from a range of sources. To achieve this, one of the key technical challenges is to provide a *middleware infrastructure* such as the one we have created, as described in Figure 1.1. As discussed above, this provides a *management interface* that enable a learner to control which sources of information should be incorporated into the model. Essentially, this has two parts: selecting and activating the right aggregator *app*; linking it to the right part of the model. For example, suppose Alice (from our scenario) took a course in anatomy, using both a learner management system and the *Anatomy Tutor*. If these are open, in the sense that there is an API which enables programmers to access their data, our approach requires an aggregation application for each of these sources. Each accesses Alice’s data, adding it to her lifelong learner model.
1.1 Roles for lifelong learner models and technical challenges

To link a new source of data to the model, it will often be necessary to deal with two different ontologies, or schemas: one from the data sources and the other from the learner model. There has been considerable research on automated approaches for such harmonization. These are particularly important where there is no user in the loop. In the case of the lifelong learner model, the learner may be able to play a role also if needed. They may also want to have control over this aspect of their model.

Consider an example of learning a complex skill, such as programming. One aspect is knowledge of loops. The term *loops* has an agreed meaning and it is easy to map this to other synonyms, such as *repetition* and *iteration*. This can be automated. By contrast, for the case of Alice's long term fitness goals, Alice may choose what she considers important as fitness goals and how the various data available should be associated with them. So the system should allow her to establish her model to match her conceptualization of this part of the model.

*Sharing the learner model with people*

With the lifelong learner modeling holding an aggregation of useful information about the learner, it becomes critical that the learner be able to manage its use. This is the complex matter of privacy, and the associated matter of release of information so that it can be useful (Palen and Dourish, 2003). As indicated in the table, the challenge is to create interfaces that can make it easy for the learner to achieve this. A suitable underlying representation and a focus on simplifying the learner model are likely to facilitate the creation of such interfaces.

There has been some exploration of the interfaces for Open Learner Models that enable a learner to control sharing of their model (Bull et al., 2007a). A lab-based evaluation study indicated that there were students who wanted to view the models of their peers and, reciprocally, there were enough students willing to share their models to make this facility useful. To support flexible sharing of arbitrary parts of the lifelong learner model, a key challenge will be the creation of effective user interfaces for the learner to do this safely and easily.

There are other cases where it would be valuable to provide OLM interfaces that share the learner’s model with others. For example, for a child, it may be valuable to create one application to present the model to the child and another for their parents. Lee and Bull (2008) explored just this, so that parents could help their child with math homework. Similarly, Zapata-Rivera et al. (2007) explored a specialized interface for parents for standardized test results, with personalized explanations of the information.
The lifelong learner model should support a *flexible range of interpretations* of parts of the model. For example, continuing the example of modeling the learner’s knowledge of loops, consider the definition of when a learner *knows* loops. In a teaching system within a first programming course, it may be reasonable to judge a user as knowing loops if they can read and write simple code in one language (Lister et al., 2004). For a higher level student, one would expect far more, such as the ability to read and write more complex loop code, in more languages, more quickly and with fewer errors. For example, Joel Spolski, a well known commentator on software and programmers notes the value of testing potential employees on very straightforward tasks, and observing if the person can solve them very quickly and easily, reflecting significant over-learning for that level of knowledge and skill (Spolsky, 2006).

There is no simple and sound basis for defining when something is *well enough known*. For the case of the individual learner, competency assessments or mastery standards may be useful. The more usual approach in classroom contexts is to provide the learner with comparisons of their own performance against a suitable peer group. For well calibrated tests, the peer group may be based on age or educational level. More commonly, it is simply the comparison against the students enrolled in the same subject. For long term models, this may be of decaying usefulness. This is especially true if the student has barely grasped the ideas, but was able to demonstrate them adequately on the particular day of an assessment, or with assistance in a classroom or homework setting.

Consider another example of the need for flexible interpretations of the lifelong learner model. If the user returns to the teaching system after several months, or years, how should the old learner model be interpreted? Perhaps older evidence for learning should be treated as less reliable? This, too, depends on whether the learner really mastered the knowledge in the first place, whether there have been practice possibilities, whether it linked to later learning in some way and the nature of the knowledge.

A somewhat different class of flexibility in interpretation is linked to privacy. The learner may be happy for a very complete model to be available to some people but for others, they may want to restrict some classes of evidence about their learning. This could be achieved via filters on the evidence, perhaps in combination with specialized interpretations of it.
1.1 Roles for lifelong learner models and technical challenges

Reuse by different applications

Reuse is one of the major potential benefits of the lifelong learner model. Because multiple programs need to be able to use and reuse parts of the model, each must be able to interpret the model appropriately. Table 1.1 shows three technical challenges.

The user interface challenge relates to providing user control of the reuse of the model. In the discussion of Figure 1.1, we showed that the middleware infrastructure should support this, via the control interfaces so that the user can decide which apps to link to which parts of the model. As illustrated in Figure 1.1, some of these will enable applications to interact with the lifelong learner model to support learning. In the figure, this was the case for the Anatomy Tutor’s use of the learner model.

The second challenge shown in Table 1.1 relates to the tasks for the middleware. It must ensure security and manage access to the model, so that the user’s privacy requirements are met. The table also shows that active delivery of parts of the model may be needed. For example, an active learner model in conjunction with a fitness application could alert the learner automatically, for example prompting them to do exercise to meet the goals that they had set themselves.

We now consider the third technical challenge of interoperability between application and the lifelong learner model. If an application is to reuse parts of the model that were created by other applications, there needs to be a common ontology or schema for that part of the model or a mechanism for mapping between different schemas. There is also the need to take account of potentially different standards and ways to interpret the value of parts of the model. For example, a teaching system for senior programming students could reuse the model from the earlier stages. But it would need to “understand” several aspects beyond the semantics of loops: it should take account of the time since the student learnt about them, the source of the evidence about their knowledge and then it would need to interpret these. For some aspects, there may be established competency standards.

Of course, it may turn out that it is simpler not to keep most learning data. As now, if there is a need to determine what the learner knows, they could be asked to do a test. It will be important to determine what classes of learner modelling information are better addressed in this way. It seems likely that the long term learner model will be of real value for the very long term goals, such as Alice’s fitness. It may also be most useful where there are services that act automatically to help the learner achieve these goals.
Institutional use of aggregated long term learning data

This important possibility for long term learner models could provide new insights about education, within an institution, or more broadly, based on data mining the long term learner model. This is a goal for the rather new field of educational data mining, EDM (for an overview, see Romero and Ventura (2007).) There is potential value in mining individual learner models, for example, sequence mining to identify interesting and important patterns of behavior. From the learner’s perspective, these might require the learner to authorize applications to use their model. The results of the data mining could be made available to the learner, either by adding them to their model or via applications for viewing the learner model.

In terms of the way that a large organization operates, there is potential for better recognition and exploitation of the knowledge and skills of each person within the organization. Notably, it might help reduce inefficiencies due to the failure to take account of a person’s knowledge when they need to undertake new training.

As shown in Table 1.1, there are several technical challenges that need to be addressed if organizations are to make better use of long term learner models. First, is the need for effective interfaces to the models, enabling effective views of the right information, while ensuring appropriate respect for individual privacy. There is a need for middleware infrastructure that makes it easy to aggregate the learner model information and to access it as needed, in the form needed. As in many other roles, ontologies and standards have the potential to play an important role in effective management of the user models and reasoning about them. Importantly, these technical elements need to fit seamlessly into the organizational structures, a significant challenge in the large organizations that might potentially gain so much from improved understanding of staff learner models.

Summary

Across the range of roles, two the recurring technical challenges relate to user interfaces and the challenges of ontologies. These are the focus of the examples we present in the next sections.

1.2 Interfaces to learner models

In the scenario, there were several cases where Alice needed to reflect on her learning progress, by viewing her learner model. In general, there are many hidden learner models within computer systems. In the case of conventional
1.2 Interfaces to learner models

e-learning systems, there is often considerable data about the learner’s activity when interacting with the system. Much of this is unused, although it has the potential to contribute to a learner model. Within typical Learning Management Systems (such as Moodle, Sakai, Blackboard), typically, some information is made available to the learner. (For example, results of assessments are usually available, often with an indication of performance of the rest of the class.) The teacher typically has access to far more detailed information about student activity on the system. In this case, there is some limited use of learner data via the system interfaces and there is support to export data, such as marks. In addition, more recent versions of LMSs such as Moodle, enable the teacher to associate parts of the site with their subject’s learning goals. This is a first step in making current web-log data, as well as assessment results, ready to be built into a learner model. There has been some work to create useful visualizations of such data, particularly for teachers to track the class and individual activity and progress, for example Mazza and Dimitrova (2007). At the other end of the spectrum, intelligent tutoring systems, with rich learner models, typically keep them completely hidden from the learner, deep within the system.

An open learner model aims to exploit the learner model to serve several potential goals, including supporting reflection, planning, maintaining a sense of control and responsibility (Bull and Kay, 2007).

The remainder of this section discusses two examples of open learner models each representing one important class of research. The first tackles the challenge of large user models, for use over long periods of time for a large learning corpus. The second explores the role of an open learner model within a group context. In this case, learning goals include group work skills. These are representative of important generic skills, being particularly highly valued in the workplace. Importantly, such skills can only be developed over several years.

Viewing large learner models

We now describe our work towards interfaces onto large learner models. The initial motivation for this work came from the medical program at Sydney University. This was a 4-year graduate program, using Problem Based Learning (PBL), where students worked in groups on a new problem each a week. For each problem, students were presented information about a patient, often spread through the week. They needed to learn enough to conclude what to do in terms of diagnosis, additional tests and treatment. Importantly, over
the program, students needed to learn the essential foundations which were tested in a critical exam being at the end of the second year.

The nature of PBL means that students learnt many things as they tackled each problem. However, as each problem was designed with particular goals, students needed to check they had learnt these. In addition, since many areas of learning interact, students needed to revisit topics, reviewing the links with things learnt more recently. It was quite challenging for a student to be confident they were making adequate progress. This was partly because there was just so much to learn. As an aid, the students had access to a large database of multiple-choice questions which covered the syllabus and were in a similar format to their actual exams. It had over 600 learning topics, each with 10 questions. Even with this, it was hard for students to see their progress over the two years of its use.

This situation motivated our exploration of open learner model interfaces that could help with this class of problem, involving a large learning corpus, over long periods of time. We created learner models, based on evidence from the student’s performance on the multiple-choice questions. Then we needed an interface that could open these learner models, enabling students to see how well they were doing and to plan their study to fill gaps. The resulting visualization interface was VLUM (Uther and Kay, 2003) for viewing very large user models. In the first work on VLUM, the model was structured in a graph, created to have suitable levels of fan-out for effective visualization, and based on similarity of the concepts. In later work, we used online dictionary sources to automatically generate the graph based on an ontology for the domain (Apted et al., 2003) with relationships such as synonym, is-a and is-related. In the case of is-a and synonym relationships, these support inference about the model, most importantly, across granularity levels.

We now explain the particular demands for a visualisation of this type of large user model and the ways that VLUM addressed them. First, like any visualisation of large amounts of information, interfaces to large user models should support the key facilities: overview first, zoom and filter, then details on demand (Schneiderman, 1996). VLUM achieves an overview with an animated visualisation that was designed to give a quick overview of progress. It enables the student to focus on particular parts of the model (zoom). The user can control the standard required for success (filter). VLUM enables the learner to see the reason for the learner model conclusion, by providing a link to the actual questions and the student’s answers (details on demand). In addition, since the learner model is based on reasoning that must take account of correct and incorrect answers, VLUM indicated the certainty of this inference about the level of learning the student has achieved.
An example screen is shown in Figure 1.2. It displays all learning topics at once. Note that on the actual screen, it exploits every pixel, and so it is harder to appreciate in print. The overall appearance, even on the screen, has much of the text very cluttered and not readable. However, the user can see the overall dominant color and our studies demonstrate that users can also readily see extreme outliers in their model.

The reduced size printed versions of the dynamic VLUM interface are
difficult to appreciate. We conducted detailed evaluations of VLUM (Uther, 2002). In order to be able to run these with a very wide range of participants, we did not use the medical domain learner model. Instead, we created a model of movie preferences, a subject readily understood by most adults. We randomly allocated participants into four groups. The first group saw VLUM displays for 100 concepts; the other groups worked with 300, 500 or 700 concepts. Participants worked through a short online tutorial introduction and then answered questions requiring them to find concepts strongly liked, strongly disliked, with high or low certainty, and the use of different standards. We measured the correctness of answers as well as the speed (although participants were not told to work quickly). Results indicated that VLUM provides an effective way to scrutinize a large user model, with up to 700 modelled concepts. We now return to the description of its interface, making use of the full screen shown in Figure 1.2 and an enlarged part of it, close to the actual size on the screen, and easier to see in print (Figure 1.3).

If the learner is modelled as knowing a topic, it appears in green, with brighter green for better known topics. Similarly, if a topic is modelled as not known, it appears in red. At a glance, the user can see an overview of their learning progress. If the display is largely green, as in Figure 1.2, it is easy to see that this learner is modelled as knowing most of the topics. It is also easy to see when there are just a few topics in red, as in the case of the topic Effect of infection on pregnancy (role of the placenta) which appears about a fifth of the way down from the top of the screen. Figure 1.3 shows this part of the screen closer to the size in the actual interface.

In VLUM, the display always has one focus topic. This is presented in a larger font and more space around it than any other topic. VLUM maintains a structure over the learner model, representing which concepts are most closely related to each other. The topics that are closest to the focus topic are presented in smaller font (and less space) than the focus, but larger than others. This approach has the effect of making the focus topic and those most closely related to it the most visible. At any stage, the learner can click on any concept to make it the focus. The Action label at the top of the display offers a search, enabling the learner to explore parts of the model they cannot immediately see on the display.

In Figure 1.2 the focus topic is Effect of infection on pregnancy (role of the placenta) and it is bright red, indicating lack of knowledge. At the bottom of the display, VLUM shows more details for the topic under the user’s mouse, in this case the focus concept. This shows that the actual assessed knowledge level (0.07). This makes it easy for a learner to slide the mouse down the
1.2 Interfaces to learner models

The learner model makes conclusions about the learner’s knowledge of a topic, on a gradient. This can be quantized for presentation in VLUM, for example on a 5, 7 or more step scale, from very well known (bright green) down to unknown (bright red). Any such conclusion is uncertain for several reasons: the learner may have done very few questions on a topic; they may have made a slip; or a correct guess; they may have done many questions on a topic, some correctly and others not. An important design goal for VLUM was to enable the learner to easily see how certain the learning assessment was.

VLUM shows the certainty using horizontal position. So, for example, a scored based on multiple consistently correct answers would appear in bright green at the left. In Figure 1.3 we see that Effect of infection on pregnancy (role of the placenta) is further right than Host responses to viral infection because the learner model has higher certainty for the latter. If the student has mixed results on questions done for a topic, with some questions for a topic correct and some wrong, VLUM displays this as yellow. So, for example, in Figure 1.3, the topic Effect of rubella on detail development is to the right, because there is no evidence about it, and it is yellow indicating that the model treats this as neither a known or unknown topic. Evaluations indicated that users were able to learn and use this information readily (Uther, 2002).

In Figure 1.4, we illustrate the effect of two learner actions. First, they clicked on Effect of virus on host cells in Figure 1.2. This is related to Effect
The second change is to the standard that the learner wants to set for themselves. In Figure 1.2 and the enlarged part of it in Figure 1.3, VLUM has the standard set at 50%. In Figure 1.4, the learner has adjusted this to 10%, using the slider just above the actual display of the model. This makes the model greener, with many more topics becoming green and light green ones becoming brighter green. Also, the formerly bright red Effect of infection on pregnancy (role of the placenta), the focus in Figure 1.2. So now, Effect of virus on host cells is the focus topic and in the largest font with most space and the former focus is smaller but still visible. In terms of font and space, the two topics have swapped.
Figure 1.5 Example of focus on *Structure of heart and great vessels*

*on pregnancy (role of the placenta)* has become a less saturated red (pink). This adjustment to the standard is particularly useful for quickly seeing the least known topics as very few remain red.

Similarly, the user could alter the standard to 90%. This makes more topics become red, or brighter red and fewer green. If the learner knew only a few topics, they would be the only green ones and would be easier to see. Of course, another reason the learner may raise the standard is that they may have a high personal standard, especially in the later period of their studies when they aim to do well on most topics. In that case, this display gives an overview of how much of the model is below their standard. In
addition, the brightest red topics are the ones they should consider first in planning their study.

Figure 1.5 shows another example, where the focus is *Structure of heart and great vessels*. In this case, various topics related to heart disease are the most visible. Note that VLUM can be configured to use the vertical dimension in a manner that meets particular learning contexts. For example, if the topics have time-stamps, provided by the teacher, reflecting when the topic was first met in the two years, this vertical dimension may be useful for the students as they may recall the approximate time that they first studied it. Such time-based landmarks have been useful in other contexts (Ringel et al., 2003). This seems a particularly useful approach for visualizing long-term learner models.

VLUM was designed to be at the left of a browser, so that the right of the screen could be used for additional information. For example, we have set it up so that when a learner selects the focus topic, they can see a list of the questions associated with it at the right. In this mode, the VLUM visualisation of the learner model can be used to choose the questions to do. In another mode, the right screen presented a list of links to the learning resources for this topic. In this case, the visualisation can be used as a navigation tool around the large collection of learning resources available for the course.

**Long term group interaction models**

We now turn to a very different example of an open learner model. This was designed to support groups of learners who work over a semester in a team project. We used variants of this form of open learner model in two contexts: a capstone software development project course where students write a substantial piece of software for an authentic client (Upton and Kay, 2009); and a postgraduate education subject where students collaboratively write reports (Kay et al., 2007, 2006).

The goal of this work was to exploit the *electronic traces* of the learner’s activity in a conventional tool to build a simple model that would support learning. In our case, our students used trac (http://trac.edgewall.org/) an open source tool that provides a set of tightly integrated media to support a group project. It was originally designed for teams of programmers. Its basic form provides three media. One helps the team define the tasks that members need to do; each task description (called a *ticket*) is associated with a higher level milestone. So, for example, suppose a team needs to find literature related to their project and submit a report summarizing it
1.2 Interfaces to learner models

by a set date. They would define a milestone for this report, due at that
date and then the team members would identify the many tasks to be split
among them, creating a ticket for each, allocating these across the team. The
second medium is a *wiki*, useful for collaborative editing of documents, such
as reports, meeting minutes and resources. The third medium, supported by
a helpful interface, is a version controlled repository. (This was not used in
the education subject.) Importantly, it is easy to create links between any
of these media. So, for example, a ticket description can include a link to
the relevant part of the wiki and repository.

As team members use such a tool over many months, their activity can be
used to create a learner model. In this sense, it is representative of a large
class of widely used online tools. If we can transform such logs of activity
into a meaningful learner model, this has the potential to support long term
learning. The most obvious role for such as model is to provide better help
and coaching about that tool for the learner, for example in basic tools such
as a word processor (Kay and Thomas, 1995; Linton and Schaefer, 2000).

We created models from trac activity for each group and created a trac
plug-in visualisation. An example is shown in Figure 1.6. This is for a real
group of 5 students, but has been anonymized. The vertical axis is time, the
bottom being the beginning of the time displayed, with one cell for each day.
Each of the 5 blocks represents one student’s model. As the legend indicates,
the leftmost part of this, in purple, represents wiki activity, the next, in
blue, is activity on the svn repository and the third in green shows ticket
activity. The intensity of the color indicates the level of activity. (Students
can customize the thresholds for the levels.) So, for example, the leftmost
student, *member1*, had quite high levels of activity on all three media on
the first day. However, on the last day shown, they had no activity at all,
indicated by the grey cells for all three media. Clicking on one of the cells
displays the details of the activity on that medium on that day. In the figure,
one can see a hand where the user clicked on the svn model for *member2*.
The details appear as the 5 lines at the right, each to one *changeset*; clicking
on the blue number within any of these presents the precise details of the
 corresponding changes in the repository.

Below each student’s daily details is a summary histogram. This has a
grey bar for the group average and a colored one for the individual’s activity
level. So, a student who does less than the average has a grey bar poking
out from their own level. A student who is doing more than average has
a pleasing purer colored bar visible beyond the grey-backed part. So, for
example, *member1* had wiki activity similar to the average, but was above
average on the other two media.
Figure 1.6 Visualisation of activity by each team member across three media: wiki, version repository and tickets defining tasks to be done.

This visualisation has proved particularly useful for meetings between the group facilitator and the students in the group. The students are asked to explain why some members are more active on different media at different times. They are asked to link this to the allocated roles of each student. Narcissus, and its predecessor, enabled teachers to see group problems early enough to help the students improve the function of the group. It also supported the mentor in working with students as they tried to remedy problems in the group management and operation.

An important goal for this work was to exploit the electronic traces from the activity of team members to create a form of learner model that would enable each member of the team to see their own activity, as a whole, and in relation to other team members. An unintended but valuable side-effect is that the visualisation of the learner activity model also serves as a new way...
to navigate around a very complex information space composed of many versioned wiki pages, many version-controlled files and tickets.

Our experience indicates that it is quite important that a facilitator helps groups in reviewing the Narcissus models. We briefly considered the possibility of creating a mechanism for sending automated advice; but we rejected this as infeasible. This is partly due to the complexity of the tasks undertaken by the groups, in building a substantial piece of software over a full semester. It also follows from the fact that each project is different, as each group has their own, real client. This situation would be even worse for general use of a tool like Narcissus, in conjunction with any of the range of group projects that trac is typically used for.

Perhaps even more important is that the actual nature of the ideal contribution is very difficult to formulate. For example, the team leader has very different roles from the person whose main role is as a programmer or a tester. Importantly, groups were asked to write a team contract and as this was under their own control, the ideal behavior should be defined by that. These aspects mitigate against the possibility of a specialized advisory tool. So, we opted for the mirroring end of the spectrum that runs from mirroring to meta-cognitive tools to guiding (Soller, 2005). We have explored use of data mining to identify patterns that are associated with effective performance of a particular role and did find quite promising ones for distinguishing whether the leader is performing that role effectively (Perera et al., 2008). Certainly, our experience is consistent with that of Jermann and Dillenbourg Jermann and Dillenbourg (2008) in the need for supporting learners in judging the appropriateness of the contribution levels of each member of the group.

1.3 Ontologies for lifelong learner modeling

This is the second recurring technical challenges of long term modeling. It is key for aggregation of information from diverse sources and for reuse of the learner model by different applications. A learner model must be defined in terms of an ontology and the particular challenge for the lifelong learner model is that it must be able to make use of information about the learner where that is framed in terms of the ontologies of the sources. So, for example, if the learner makes use of five different math teaching systems, the lifelong learner model must be able operate in terms of the ontologies or schemas of each of these. There is a huge body of research on ontologies and the harmonization of different ontologies. In this section, we illustrate a lightweight, highly flexible approach to these problems.
1.3.1 Ontologies built from specialized glossaries and dictionaries

The work we now describe was motivated by several potential benefits that an ontology could provide, even for short term learner modeling within a single application. Although we had not considered long term learner modeling at that time, our motivations and approach turn out to be useful for it, as we now discuss.

One valuable role for an ontology is to provide a structure over a learner model so that it can be presented more effectively in an open learner model. For the case of VLUM, we wanted this to structure the graph so that it would present the concepts most closely related to the focus concept. In this case, we need an ontology with the right fanout from each concept, so that there are neither too few nor too many concepts that were modelled as closely related to any focus.

The enhanced version of VLUM was called SIV (Scrubtable Inference Viewer) (Kay and Lum, 2005), its name reflecting another important role for an ontology in supporting inference about the learner model. In particular, we found that it was natural to tag each small learning task with very fine-grained concepts. For example, consider the case of an exercise designed to help students learn about pointers for string manipulation in C. It is natural to tag this task with this fine-grained concept from the ontology. But, this is part of the coarser-grained, broader topic, pointers in C. Suppose a student wants to know how they are doing on this broader topic, but all the data sources for the learner model are for the finer-grained parts of the model. Then, an ontology could support reasoning that would enable a system to model at both fine and coarse grain. Similarly, when students undertake a large assignment which involves many elements of the course, this must be coded at a coarse grain. An ontology can support (uncertain) reasoning about the learner’s finer grained knowledge. Such inference can be valuable for the task of building comprehensive learner models and for presenting them as OLMs.

We created a flexible mechanism for building ontologies for learner modeling. Importantly, we automated the process of building a basic ontology and also enabled a teacher to readily augment the ontology with their own concepts. For example, in a subject on the C programming language and unix, we introduced the notion of Core Material where this included all the concepts required to earn a bare pass. In the context of this subject, which many students found difficult but were required to pass, the teacher wanted struggling students to be easily able to determine where to focus.
1.3 Ontologies for lifelong learner modeling

The learner model was populated with evidence from an on-line learning system that provided many small learning activities for C programming. All were tagged with fine-grained concepts from the course ontology. Yet, the SIV open learner model could use the ontology to infer the learner’s progress at the high level. The coarse-grained concept *Core Material* has child concepts (specialization and part-of) that include about two-thirds of all modelled concepts. The visualisation of the *Core Material* gives the learner an indication of their knowledge at that level, even though there is no direct evidence about learning at this coarse-grain. SIV infers the value for this part of the model, based on the learner’s performance on the full hierarchy of concepts for that part of the subject. In principle, SIV could make use of any ontology and we used it with several, for teaching different aspects of programming and user interface design (Lum, 2007).

While there are established approaches to building ontologies and many existing ontologies (Noy and McGuinness, 2001), these are based on widely accepted formal approaches. They lack key properties that we wanted. First, we wanted the ontology to be *easily extensible* by the teacher of a subject. As we have noted, this was important for concepts that were important to the teacher, but not the general area of knowledge. This was the case for the term *Core Material*, as discussed above. This concept served an important purpose for this subject and it was just this type of concept that we wanted teachers to be able to add to the learner model ontology. This is a problem if one uses formal ontologies since they are not readily understood or managed by a classroom teacher. A second reason for rejecting widespread formal approaches was that we wanted to create *scrutable learner models*, meaning that the learner should be able to scrutinize it to determine what it means and how it reasons about their learning. One important aspect of this is to explain the ontological reasoning. This should be feasible with formal ontologies but would involve considerable research to ensure that explanations were at the right level for the learners.

Our approach to building the ontology was to create a tool, called MECUREO, that analyzes a dictionary or glossary to create the ontology (Apted and Kay, 2004). In some of our work, we used substantial online dictionaries. For the C programming subject, we created a glossary that served both for building the ontology and as a resource for the students. Part of it is shown in Figure 1.7.

Within the learning system used for this subject, the teacher can add new concepts to the ontology at any time. The interface for this is shown in Fig-

---

Subject Glossary

<table>
<thead>
<tr>
<th>Category/Concept Name</th>
<th>Category/Concept Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>The essential concepts that students must learn to pass SOFT2130. These are &quot;Similar concepts in both C and Java&quot;, &quot;Pointers&quot;, &quot;Pointers with Strings&quot;, &quot;Pointers with Arrays&quot; and &quot;Dynamic Data Structures&quot;</td>
</tr>
<tr>
<td>Core/Similar concepts in both C and Java</td>
<td>Similar ideas shared between C and Java</td>
</tr>
<tr>
<td>Control Flow</td>
<td>All the elements of the programming language that affect the order of execution of the statement in a program. For both Java and C, these are the selections structures, if, if-else and switch, the looping or iteration structures, for, while and do-while as well as the basic default flow means that statements are executed in the order they appear on the page.</td>
</tr>
<tr>
<td>Function Arguments</td>
<td>A value or reference passed to a function or procedure or subroutine or command or program by the caller.</td>
</tr>
<tr>
<td>Arrays</td>
<td>A collection of identically typed data items distinguished by their indices. The number of dimensions an array depends on the language but is usually unlimited.</td>
</tr>
<tr>
<td>Scope</td>
<td>The scope of an identifier is the region of a program source within which it represents a certain thing. For both C and Java, the basic scope rules are much the same. One of the important things that you will learn in this course is about the runtime storage of data: once you understand this, you will appreciate why scope operates as it does. So, for example, you will realise why the code in one function cannot access data that is declared within another.</td>
</tr>
</tbody>
</table>

Figure 1.7 Subject glossary used to create lightweight ontology

Figure 1.8. At the left, the system shows the existing ontology. At the top is the concept *Similar concepts in both C and Java*, followed by several concepts. The reason the teacher chose this structure is that the students in this subject should have completed two programming courses that used Java. So, both the teacher and the students would find this conceptualization of the domain meaningful and useful. The middle of the screen provides facilities to *Add a concept* to the ontology. To do this, the teacher needs to select the concept category from the popup menu (currently on SOFT2130/Core, the course name being SOFT2130. The next box is for the new concept name followed by the definition as it will appear in the glossary. This interface contributes to the definition of the ontology in two ways. First, there is the structure that comes from the hierarchy explicitly defined by the teacher, such as SOFT2130/Core. The second comes from analysis of the dictionary definitions, defining links between all concepts defined by the teacher.

When the teacher creates new learning tasks, they select from the available concepts in the ontology, linking them to the task. The interface for this is shown in Figure 1.9. In this case, the teacher has linked four concepts to the task. In this system, these will each be linked to elements of the task in later stages, so that as the student does the task, their actions provide evidence for the learner modeling.
1.3 Ontologies for lifelong learner modeling

Figure 1.8 Defining a new concept

Figure 1.9 Authoring interface for linking concepts from the ontology to learning tasks
This approach enables a teacher to easily create an ontology based on concepts that are natural for them. When we applied this approach to a user interface subject, we used the SIV visualisation of the ontology as part of the interface aiding in the tagging of each learning object with the relevant parts of the ontology (Apted et al., 2004). In that case, we aggregated data for the learner model from different sources (Kay and Lum, 2005): very fine grained evidence from each of slides of the online audio lectures for the subject and very coarse grained evidence from the larger assessments, such as are commonly stored in LMSs.

The approach can very gracefully handle problems of unstable or changing vocabulary. The teacher simply creates new, local dictionary definitions, where these introduce the new terms and, in normal English, explain that it is the same as, or similar to, other terms from the ontology. These are then available to the students in a class. We used this in the case of our HCI course where some terms are not stable and those used the dictionary were not the ones in our textbook. Simply adding a definition, means that the ontology could readily reason in terms of both the terms from the on-line dictionary and those used in the subject. An additional benefit is that the class glossary could explain this.

1.4 Discussion and conclusions
At some level, all online learning tools can collect data about the learning, at varying levels of accessibility for the learner, or others supporting their learning. At one end of spectrum, there are some emerging teaching systems that maintain a valuable learner model, perhaps with sophisticated interfaces onto it for use by the learner and possibly others who support them. At the other end, there are merely digital foot-prints in various logs of the user activity. At present, we make little use of most learning data even in the short term and it is rarely pressed into service supporting long term learning. Typically, it resides in collections of silos, one for each particular system or part of it. The lifelong learner model has the potential to change this situation, making more of this information available in a form that can support learning.

This chapter has explore the new roles that a lifelong learner model might play for open learning modeling, for aggregating information from diverse sources, enabling the sharing of it with other people under the learner’s control, supporting flexible interpretation of the model, reuse of it by applications that the learner uses and institutional use of the models of many learners to improve understanding of their learning. We have also identi-
fied key technical challenges that need to be overcome if the lifelong learner model is to fulfil these goals. These take the form of several challenging issues for creating effective interfaces for control over the lifelong learner model and effective views of it, ontological challenges to support effective aggregation, interpretation and interoperability. And we have identified the need for an effective infrastructure and given an overview of one that we have created.

We began this chapter with a scenario that provided examples of some important classes of very long term learning goals; those associated with learning related to long term goals such as fitness; and those associated with development of expertise. Such learning goals, being so long term, call for the sort of lifelong learner modeling framework we have described, with diverse sources of evidence about the learner and the potential for learning benefits derived from using many teaching and learning systems. Of course, shorter term learning goals could also make use of the lifelong learner model, reusing relevant parts of it. This chapter has also focused on the potential role of open learner models, in association with long term learner models. We have described our exploration of interfaces for large scale, long term learner models and lightweight ontological approaches that explore a flexible approach to the challenges of representation of the model. Based on our analysis of the technical challenges of lifelong learner modeling, our work on these two aspects, user interfaces and ontologies, is part of a foundation for supporting lifelong learner modeling.

Acknowledgments
This research was supported by Australian Research Council Projects DP0774532, DP0877665. We are grateful for the insightful comments from the reviewers. Some of these are reflected in important new material that contributes significantly to the paper.
References


Bull, S., Mabbott, A., Gardner, P., Jackson, T., Lancaster, M.J., Quigley, S., and


Uther, J., and Kay, J. 2003. VIUM, a Web-Based Visualisation of Large User

