Active Open Learner Modelling

Collene Hansen and Gord McCalla
ARIES Lab, University of Saskatchewan,
57 Campus Drive, Saskatoon, Saskatchewan, Canada S7N 5A9

Abstract. By opening the learner model to both the learner and other peers within an e-learning system, the learner is able to reflect on the contents of the model. Information in the model depends on the context in which it is generated. Current systems translate an *existing* model to fit the context. The active open user modelling approach presented in this paper describes a method of *generating* the model to the specific user and purpose. The main advantage of this approach is that exactly the right information is generated to suit the context and pedagogical goals of the system. As the interaction with the student proceeds, the model is regenerated at each interaction, allowing it to stay in tune with the context. This same approach can be applied to actively open models of others in the system for many purposes, including comparative reflection.

1 Introduction

While most systems keep the learner model hidden from the user, recent research has explored the benefits of opening the model up to the users of the system, so-called open learner modelling [1]. The primary reason to open the model is to encourage reflection as a learning process [2]. As the student views and/or interacts with the information within the model, he or she reflects on his or her model's characteristics as described by the system, thus gaining a greater understanding of the domain, his or her current beliefs, and performance within the system [3].

Reflection need not consist solely of viewing the model, but may also involve interaction with the model. In this approach, the student can not only view the information contained within the model, but can influence or change parts of the model or modelling process [2]. This allows the student to determine what the system has discovered about him or her and challenge parts of the model if seen as inaccurate [4]. In addition, because the user is able to view and manipulate the information stored within the model [5], the user reflects on the learning process and is perhaps motivated towards the goals presented by the system [6].

A related but more complex issue is the idea of opening up the learner model to the scrutiny of others: peers, tutors, and teachers. This requires a different approach, as users of a public model have different purposes than a user viewing his or her own model. Allowing the user to access other models provides many advantages. By comparing their own characteristics to others in a course, the learner may gain a greater understanding of his or her problems in the domain by directly comparing results to those of an expert or average student, as in Bull and Nghiem [3]. This provides the opportunity for a student to compare their progress against that of their peers. Viewing a potential helper's characteristics can aid in choosing the best helper from a list [7]. Viewing a learner's model when giving help allows a helper to tailor the help response

to the specific needs of the request. Of concern here is to provide useful information for the viewer's needs, designed specifically to what he or she wants to discover from the model, or to provide the opportunity for the user to find and choose the characteristics he or she would find the most helpful.

Current open modelling research has either limited the characteristics viewed by the student in the model [3] or provided all the information for the user to sort through alone [8]. Instead of having one large model displayed, with many details, smaller portions based on the specific purpose the user has to view the model can be displayed instead. This narrows down the information to only what is required, without limiting the usefulness of the system. The *active modelling approach* [9] can effectively be applied in this case.

Most user modelling systems maintain a data structure which stores the information in the model but does not specify how to interpret the information or decide how the information is to be used. For example, a stored model may contain calculated fields such as "knowledge levels", which are updated periodically as a student interacts with the system. The calculated field is a compilation of all previous interactions with the student.

However, active modelling views learner modelling as a process, where the model is computed for just-in-time delivery when a particular need arises. There is not one large model, but many fragmented models or pieces of raw data that are retrieved, integrated, and interpreted according to the task and context. So, an active model can be thought of as a function:

$$\text{learnerModel}(a, L, p, R) \quad (1)$$

with the following parameters, the context for the modelling:

- a=the agent (or process) performing the modelling
- L="other" learners in the system (which can include peers, tutors, and instructors)
- p=the purpose for calculating the model
- R=resources available to the agent to perform the calculation, such as time or memory.

In the active approach, the focus is on developing clichés that capture typical learner modelling computations. Early research into the nature of such clichés suggests that they are often oriented around the various purposes (p) that underlie the learner modelling computations [10].

We would like to apply the active approach to problems in open modelling. Then the natural question is: how do you open the model when there is no model? However, considering that this is the active approach, this should be rephrased as: when the model needs to be opened, how do you compute it? Users and purposes provide context constraints which determine what information the user may be after when viewing the model; this will be explained in the remainder of the paper.

2 Active Open Learner Modelling for Reflection

When combining the open and active learner modelling paradigms, a key question must be asked: What does it mean to open up the model if the model does not exist? In particular, in reflection with interaction between the student and the system, how do you open up the model at each stage in the interaction if there is no model? What is unique

to the purely active open approach is that technically the model is fully recomputed from scratch with *every* interaction as the context clarifies what information is needed about the learner at each stage. In this paper, we focus not on the interaction, but on the issue of how context is important at a given stage to determine what learner modelling information is needed at that stage. Specifically, we focus on the initial model (based on the first interaction with the user).

2.1 Using context to establish content

Recall from the introduction that context is a function of the agent performing the modelling, other learners, a purpose, and the available resources. We would like to show how two factors (purposes and learners) affect the context and content of the model.

2.1.1 Purposes

The purpose seems to be the driving parameter in active modelling [10]. The active model must be calculated as the purpose requires, and the purpose ultimately determines what to open up to the learner and how. However, the user may have many different purposes for viewing the information in the model. For example, a learner may be reflecting on his or her standing in a course or alternatively trying to choose the best helper from a presented list. Even within these broad categories, there may be further refined purposes, such as comparing results in a quiz to an expert or comparing results in a quiz to a particular learner. Then depending on the purpose, there is some information that is important and other information that is unimportant. For example, consider these scenarios:

1. The learner is comparing results in a quiz to an expert (or an average student). In this case, the learner may be interested in knowledge levels and comparisons with his or her own. Background information or other details may become less important.
2. The learner is comparing results to a particular learner. In this case, the background information may become more important, as the student may be trying to find reasons why the other learner is performing better or worse compared to him or her.

2.1.2 Learners

Another contextual element is the users involved in the modelling. In traditional ITS's, there is just one learner, and the system is adapting to that specific user. Of course, it is becoming increasingly important that a system also supports more than one learner—perhaps an entire virtual community (e.g. I-Help [11]). Thus these other learners have an increasing effect on the learner model.

In the active approach there are three types of users important to the context: who is doing the modelling (a in Equation 1), who is being modelled (a, L), and who is supplying the information (L). The person doing the modelling is the most important, as he or she determines the purpose for calculating the model. So the information should be generated/displayed based on this user's characteristics which again vary according to the purpose.

Moreover, the user could be playing a number of different roles, such as a learner, a teaching assistant (TA), or perhaps an instructor. This role, if determined, helps to discover the learner's purpose and provides context to open the model. Each user plays

different roles at different times, and may be playing more than one role at once. However, for simplicity, we will assume the user plays only one role at a time.

The relationship that exists between the user doing the modelling and the user being modelled also provides added constraints—who I am and who you are determines what sort of information I need to see about your model or have access to. For example, there may be two instructors: one teaching the user's current course and another instructor in the department. The instructor of the course should have access to the information about the learner related to the course, but the other instructor should not. In this case, the relationship that exists between the instructor and the learner provides constraints on what sort of information is permitted.

2.2 A suggested approach to active open modelling

Our approach to dealing with these issues is to define taxonomies of purpose clichés and standard user roles, then to use these purpose clichés and user roles to carry out typical open active learner modelling computations. Recent work in purpose-based active modelling by Niu et al [10] has described a purpose taxonomy based on generalization and specialization of purposes, which we expand on here.

Each user is playing a role during the construction of the model. Based on this role alone, a model could probably now be constructed. However, other information is present in the relationships that exist between two or more learners playing different roles in the computation of the model. So from these relationships, a default open model specific to a particular purpose can be developed. In the case of a purpose involving only one learner, relationships with other learners may still exist. For example, other learners may provide assessment of a learner's knowledge or become a base for comparison during reflection. In this situation, the learner's own characteristics and role constrain the information generated in the model.

During each instantiation of the model (i.e. during each interaction with the learner), the user's role and purpose are determined, and slots in a template are filled to match the purpose clichés. The generated information is then filtered or polished in a way specific to the purpose and the user (i.e. by generating a model to determine how to display the information to the learner). This is roughly analogous to content planning in an ITS [12], while the polished backend is analogous to delivery planning. The example in Section 3 describes this process in more detail.

3 Active Open Example

This example and the discussion that follows have been modelled loosely on the I-Help one-on-one system as described in [9] and [11]. Assume that at a given stage in the interaction between the learner and the system, the system decides that a reflective model best suits the learner's needs. We will show how context is used to generate the initial model to the user. Of course (as mentioned previously), the generation of the learner model will be repeated at each interaction. Again, the key is that the *context* constrains and actually produces the model.

The example is deliberately simplistic to demonstrate the procedure. However, the method can be extended to include more complex system reasoning, such as finding misconceptions in domain areas. In fact, it is expected that through interactions with the student, new and unexpected paths may arise as a refined purpose reveals a misconception in a completely different area than the original purpose.

In the example, "viewer" refers to the person constructing the learner model (either of himself/herself or of another student). "Viewee" refers to the student whose model is being generated and displayed to the viewer.

3.1 The example

Consider a help request system for a first year Computer Science course with Java programming. The learner models are open to both the learners being modelled (for reflection) and others in the system (comparative reflection). There may be other learners involved—another peer during comparison, other learners from whom the data was received, tutors, or instructors.

Two reflective purposes have been identified for this example: reflection on one's own model, and comparative reflection. Each of these has been broken down into two sub-purposes: reflection on the domain (i.e. knowledge levels) and social reflection (i.e. helpfulness, etc.), then particular instantiations provided at the leaf level. This is summarized in the purpose taxonomy in Figure 1.

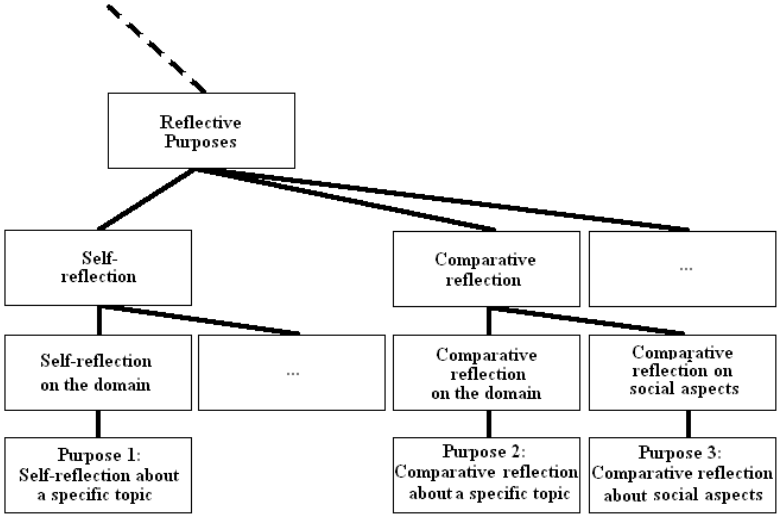

Figure 1: Purpose taxonomy for the example

The user taxonomy in Figure 2 describes each of the roles which are possible in the purpose hierarchy. Three roles have been identified for the example: learner, average learner, and expert learner. A "learner" is any student user in the system (who can have a varying expertise level), an "average learner" refers to a compiled average among all the students in the course or system, and an "expert learner" refers to an instructor or an answer source. Some roles do not apply to certain purposes; there is no "expert" for peer comparison.

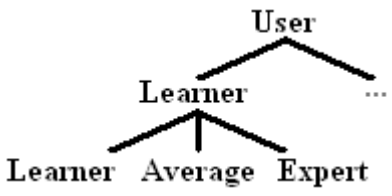

Figure 2: User taxonomy for the example

In the following tables, we instantiate several purposes with interactions between the purpose and the roles the users play. Each table describes a different situation. Consider the two learners involved in the purpose as X and Y and the topic as Z. While the purpose number is assigned from the purpose hierarchy, the complete instantiation depends on characteristics and situations derived from the users' roles. In the information displayed row of each table, all *possible* information that could be generated for each particular purpose is listed, but only the information in bold font is *actually* calculated and eventually displayed. Notice that some pieces of information are not generated for the purpose, but each piece of information is used for at least one purpose in the system. This information is not stored in one monolithic model, but is calculated for the user's needs based on the user's purpose for obtaining the information. Therefore, different information may be generated for slightly different needs, though the sources of information may be the same.

In the self reflecting example in Table 1, the user has requested his or her own model related to a specific topic in the domain, which is in this case is recursion in Java. Not all of the information that could be generated for the model is provided—only that which the user is particularly interested in according to his or her purpose for viewing the model. For example, the "Quiz 2 results" are provided, as these are related to recursion. However, the Quiz 1 results are not provided, as these are not related to the purpose the user has to generate the model.

Table 1: Self reflecting on domain example

Purpose 1: Self reflects on own model about the course		
Learner (X):	Self	
Other Learners (Y):	Self	
Roles:	Self is Viewer as Learner, Self is Viewee as Learner	
Action:	Self views own model on topic Z	
Topic (Z):	Recursion	
Resource Constraints:	Real time	
Information Displayed to X about Y:	Knows if-else statements (Java) Knows loops (Java) **Knows recursion (Java)** **Marks for assignment 1 (includes recursion)**	Marks for assignment 2 Quiz 1 results (related to loops) **Quiz 2 results (related to recursion)** Education background

In Table 2, a learner (the viewer) is comparing his or her own domain knowledge to that of another learner (the viewee). The viewee's education background is included as part of the information required by the viewer, to provide background information to the viewer while he or she is looking at the quiz results or knowledge of recursion. For example, the learner could view the other learner's quiz results and ask "why did the other person do better than I did?" Providing the other's background education may provide some of these answers: e.g. "this person took a previous course in computer programming".

Table 3 has the same purpose as in Table 2, but provides a comparison between the learner and an expert learner, who in this case provides the answers to the quiz. So, the questions and the answers are provided to the student to determine where he or she went wrong with the quiz. Notice in this case that the view of the information changes slightly for the needs of the student—in Table 2, only the results were required, but now the exact responses and the expert's answers are needed.

Table 4 again compares the model to another learner, but instead the learner can see how an average learner handled the quiz. This is to provide the learner with a

comparison base with the average learner in the class with respect to each of the quiz questions.

Table 2: Learner comparison (domain) example

Purpose 2: Self compares model to other about the course		
Learner (X):	Self	
Other Learners (Y):	Self, other Learner	
Roles:	Self is Viewer as Learner, Other is Viewee as Learner	
Action:	Self views other's model and compares to own on topic Z	
Topic (Z):	Recursion	
Resource Constraints:	Real time	
Information Displayed to X about X	Knows if-else statements (Java) Knows loops (Java) **Knows recursion (Java)** Marks for assignment 1	Marks for assignment 2 Quiz 1 results (related to loops) **Quiz 2 results (related to recursion)** Education background
Information Displayed to X about Y:	Knows if-else statements (Java) Knows loops (Java) **Knows recursion (Java)** Marks for assignment 1	Marks for assignment 2 Quiz 1 results (related to loops) **Quiz 2 results (related to recursion)** **Education background**

Table 3: Expert comparison (domain) example

Purpose 2: Self compares model to other about the course		
Learner (X):	Self	
Other Learners (Y):	Self, expert	
Roles:	Self is Viewer as Learner, Other is Viewee as Expert	
Action:	Self views expert's model and compares to own on topic Z	
Topic (Z):	Recursion	
Resource Constraints:	Real time	
Information Displayed to X about X:	Knows if-else statements (Java) Knows loops (Java) Knows recursion (Java) Marks for assignment 1 (includes recursion)	Marks for assignment 2 Quiz 1 results (related to loops) **Quiz 2 *responses* (related to recursion)** Education background
Information Displayed to X about Y:	Knows if-else statements (Java) Knows loops (Java) Knows recursion (Java) Marks for assignment 1 (includes recursion)	Marks for assignment 2 Quiz 1 results (related to loops) **Quiz 2 *answers* (related to recursion)** Education background

Table 4: Average learner comparison (domain) example

Purpose 2: Self compares model to other about the course		
Learner (X):	Self	
Other Learners (Y):	Self, expert	
Roles:	Self is Viewer as Learner, Other is Viewee as Average Learner	
Action:	Self views average learner's model and compares to own on topic Z	
Topic (Z):	Recursion	
Resource Constraints:	Real time	
Information Displayed to X about X:	Knows if-else statements (Java) Knows loops (Java) **Knows recursion (Java)** Marks for assignment 1 (includes recursion)	Marks for assignment 2 Quiz 1 results (related to loops) **Quiz 2 *average results/responses* (related to recursion)** Education background
Information Displayed to X about Y:	Knows if-else statements (Java) Knows loops (Java)	Marks for assignment 2 Quiz 1 results (related to loops)

	Knows recursion (Java) Marks for assignment 1 (includes recursion)	**Quiz 2 *average results/responses* (related to recursion)** Education background

Tables 5 and 6 provide a similar analysis for the purpose of comparing a learner's social characteristics with others in the system. Information such as cognitive style, helpfulness, etc. is presented to the learner. This may be useful for learners in finding and receiving help or determining how much one should charge for help (as in I-Help [11]).

Table 5: Learner comparison (social) example

Purpose 3: Self compares model to other about social patterns		
Learner (X):	Self	
Other Learners (Y):	Self, expert	
Roles:	Self is Viewer as Learner, Other is Viewee as Learner	
Action:	Self views other learner's model and compares to own	
Resource Constraints:	Real time	
Information Displayed to X about X:	**Eagerness to be contacted** **Cognitive style** **Helpfulness**	**Online statistics** **College/major**
Information Displayed to X about Y:	**Eagerness to be contacted** **Cognitive style** **Helpfulness**	**Online statistics** **College/major**

Table 6: Average comparison (social) example

Purpose 3: Self compares model to other about social patterns		
Learner (X):	Self	
Other Learners (Y):	Self, expert	
Roles:	Self is Viewer as Learner, Other is Viewee as Learner	
Action:	Self views other learner's model and compares to own	
Resource Constraints:	Real time	
Information Displayed to X about X:	**Eagerness to be contacted** **Cognitive style** **Helpfulness**	Online statistics College/major
Information Displayed to X about Y:	**Eagerness to be contacted** **Cognitive style** **Helpfulness**	Online statistics College/major

3.2 Discussion

Notice in the example that how each of the slots in the table was filled determined what was computed for whom. The purpose was filled with three different roles of users: learner, average learner, and an expert learner. The roles, relationships, and the actions determined what information would be generated for the model. So, it becomes clear that the purposes and users involved are crucial to what needs to be modelled. These constrain the information to generate.

While the focus in the example has been on reflection, there can be other roles included in the system. For example, a help system may include different roles: a learner as a helper, a teaching assistant as a helper, etc. These roles, while not explicitly aimed at reflection, can not help but have some reflection occur as a side effect.

4 Conclusion

There are several advantages to the active open modelling approach: ease of incorporating external data, ensuring the information is updated when required, eliminating wasted time maintaining the model, and minimizing storage requirements. Information from external sources can reduce the storage requirements in a system, yet can still be easily obtained when required. This is especially important when the information is only required for one or two purposes. Because the information is only computed when it is actually required, there is no need to periodically update the models to ensure they are accurate. There is also an opportunity for more precise, targeted modelling. Self suggests: only diagnose what you can treat [13]. The information can be calculated and interpreted in the exact way it is needed as opposed to calculating it once and then needing to integrate it in context in a separate calculation.

The future direction for this work includes developing a sample system first for simulated users, to determine possible purposes and roles, then testing the system with real users. Implementing the approach in a real system leads to a number of interesting research questions—how can purposes be determined? Can all purposes be determined? Identifying all the roles and relationships may become a large problem, especially if the system is required to be as general as possible across a range of domains.

As well, content and delivery issues provide an area of research. This paper has focused on which information to generate, but the information also must be delivered to the user. The information may be delivered as a chart, as text, or some other representation based on characteristics of the user and the purpose for the modelling. Again, because no model is stored, delivery planning must be carried out just in time before displaying the model. The question becomes what part of the delivery can be derived from the purpose, and can the content and the delivery be separated?

By far the most interesting aspect of this paradigm is its focus on the end-use of the data. Opening something up usually assumes that it exists already—this is not the case in an active open approach. There is no information without a purpose, no model without a context. This paper has shown how differing purposes, roles and relationships can constrain the information generated and determine what to generate for the user. This suggests that the information itself is not the real cause for worry in an open system, but the use (or sometimes misuse) of the information. This may be especially true if some information is combined together for unknown or notorious purposes and inaccurate or inappropriate conclusions drawn.

Privacy and freedom of information laws introduce interesting problems in an active open system. In the real world, freedom of information laws say the information must be open to owners of the data (i.e. the people being modelled). However, these laws apply simply to raw data or rows in a database table, without considering the use or purposes which use the data (see [14] for current research in privacy and databases). Any conclusions that are drawn from the data are usually private to the owner of the system. An active open system supplies enhanced protection for the user, as the data, conclusions drawn from the data, and possibly the reasoning that lead to the conclusions can all be provided.

The active open paradigm can potentially be applied to other distributed environments outside ITS's, particularly where freedom of information laws apply or privacy concerns are present. For example, in a government database, where many different details are stored about citizens, users may be curious as to how the information is being used and who is gaining access to it. How can the citizen make sense of the entire model and the purposes involved? Often the information is not

even generated until the purpose is requested. If a citizen is able to select a purpose from a list of all possible uses, he or she will then be able to determine if the model generated is accurate, both implicitly and explicitly.

References

1. Bull, S., and Pain, H. 'Did I say what I think I said and do you agree with me?': Inspecting and Questioning the Student Model. In Greer, J. (ed), Proceedings of World Conference on Artificial Intelligence and Education (ACCE). Charlottesville, VA, pages 501-508, 1995.
2. Bull, S. See Yourself Write: A Simple Student Model to Make Students Think. In Jameson, A., Paris, C., and Tasso, C., editors, User Modelling (Proceedings from 6th International Conference, UM 1997), pages 315-326.
3. Bull, S., and Nghiem, T. Helping Learners to Understand Themselves with a Learner Model Open to Students, Peers, and Instructors. In Brna, P., and Dimitrova, V., Proceedings of Workshop on Individual and Group Modeling Methods that Help Learners Understand Themselves, International Conference on Intelligent Tutoring Systems 2002, page 5-13.
4. Vassileva, J., Greer, J., McCalla, G. Openness and Disclosure in Multi-Agent Learner Models. In Morales, R., editor, Workshop on Open, Interactive, and Other Overt Approaches to Learner Modeling (Proceedings from 9th International Conference, AIED 1999), pages 43-49.
5. Morales, R., Pain, H., Conlon, T. From Behaviour to Understandable Presentations of Learner Models: A Case Study. In Morales, R., editor, Workshop on Open, Interactive, and Other Overt Approaches to Learner Modeling (Proceedings from 9th International Conference, AIED 1999), pages 15-24.
6. Dimitrova, V., Self, J., and Brna, P. STyLE-OLM--An Interactive Tool in a Terminology Learning Environment. In Morales, R., editor, Workshop on Open, Interactive, and Other Overt Approaches to Learner Modeling (Proceedings from 9th International Conference, AIED 1999), pages 25-34.
7. Collins, J., Greer, J., Kumar, V., McCalla, G., Meagher, P., and Tkatch, R. Inspectable User Models for Just-In-Time Workplace Training. In Jameson, A., Paris, C., and Tasso, C., editors, User Modeling (Proceedings from 6th International Conference, UM 1997), pages 327-337.
8. Zapata-Rivera, J., and Greer, J.E. Inspecting and Visualizing Distributed Bayesian Student Models. In Gauthier, G., Frasson, C., and VanLehn, K., eds, Intelligent Tutoring Systems: 5th International Conference, ITS 2000, Montreal, Canada, June 19-23, 2000, Heidelberg, Germany: Springer-Verlag, pages 544-553.
9. McCalla, G., Vassileva, J., Greer, J, and Bull, Susan. Active Learner Modelling. In Gauthier, G., Frasson, C., and VanLehn, K., eds, Intelligent Tutoring Systems: 5th International Conference, ITS 2000, Montreal, Canada, June 19-23, 2000, Heidelberg, Germany: Springer-Verlag, pages 53-62.
10. Niu, X., McCalla, G., and Vassileva, J. Purpose-based user modeling in Multi-agent Portfolio Management System. To appear in the UM'03 User Modeling: Proceeding of the Ninth International Conference (UM'03). Johnstown, USA, June 22-26, Springer Verlag LNCS, 2003.
11. Greer, J, McCalla, G., Vassileva, J., Deters, R., Bull, S., and Kettel, L. Lessons Learned in Deploying a Multi-Agent Learning System: The I-Help Experience. Proceedings of AIED 2001, San Antonio, pages 410-421.
12. Wasson (Brecht), B. (1990) Determining the Focus of Instruction: Content Planning for Instructional Tutoring Systems, PhD Thesis, University of Saskatchewan.
13. Self, J. Formal Approaches to Student Modelling. In McCalla, G., and Greer, J. (eds). Student Modelling: the key to individualized knowledge-based instruction. Berlin: Springer-Verlag, 1994, pages 295-352.
14. Agrawal, R., Keirnan, J., Srikant, R., and Xu, Y. Hippocratic Databases. Proceedings of the 28th International Conference on Very Large Databases (VLDB), Hong Kong, China, 2002, pages 143-154.