Incremental student modelling and reflection by verified concept-mapping

Laurent CIMOLINO, Judy KAY, Amanda MILLER
School of Information Technologies
University of Sydney, Australia
{judy, amanda}@it.usyd.edu.au

Abstract. Concept mapping is a technique for externalising a conceptualisation in terms of a visual two dimensional layout which shows the propositions believed by the learner as well as other similarity and hierarchy relationships. In this paper, we describe VCM, the verified concept mapper. It is a novel form of concept mapping designed to elicit a learner's understanding, as a basis for building a learner model. We describe the display of the learner model as an aid to reflection. We report an early evaluation of the system and the ways that it has informed our understanding of the need for incremental modelling and reflection steps.

1 Introduction

Concept mapping [1] is an established technique for supporting learning on several levels. It enables the learner to externalise their understanding of a domain. This serves as a useful means of communication with the teacher and other learners[2]. At one level, the map itself might be viewed as a form of learner model. This makes concept mapping a natural and appealing approach for supporting learning via reflection on the learner model. Because it is a readily visible form of the learner's understanding, the completed concept map can be used as a trigger for learners to compare their conceptualisations, noting similarities and discussing differences, somewhat like the use of concept maps by [3].

Although that externalisation is undoubtedly important for reflection, it is important to emphasise the inherently reflective nature of the whole concept mapping enterprise. Novak [1] notes that students find concept mapping challenging because it requires them to reflect carefully on their understanding of the important concepts and their inter-relations. Since concept mapping is so challenging, it is very likely that learners will sometimes make slips and omissions in their maps.

The verified concept-mapping tool, VCM, supports classic concept mapping available in many commercial tools, such as Ygnuis, MindManager and Inspiration. However, its main use is not free-form concept mapping. It is intended for explicit mapping tasks that have been carefully defined by the teacher. For example, the teacher might provide students with learning resources to study and then ask them to construct concept maps that capture their understanding of that material. This is exactly the role described in much of Novak's initial work [1, 4, 5]). VCM takes an approach that is similar to Leake et al [2] where their system suggests suitable concepts and resources to consider.

The distinctive feature of VCM is that it was explicitly designed to capture learner models as a side-effect of the concept mapping task. With this goal in mind, we designed it to verify that model by sharing it with the student. We intend that a learner should work on their concept map. Once they are happy that it is complete, they should submit it to the system. At that point, VCM performs two main classes of action. First, it looks for features that the student should check, since they indicate errors or oddities. At this stage, the student *verifies* that they really meant to have these features. Hence its name, Verified Concept Mapper. Also at this stage, the system shows the student the user model it is about to save. This is composed of the inferred model of the learner that was derived from the map. The availability of this new learner model also serves as a basis for the student to *verify* the map, since they can look through the learner model, checking that they really do hold these inferences.

At one level, the verification phase of the VCM system serves as a mirror held up by the teacher. It allows the student to focus on the concept-mapping task as long as they need to complete it. Then, when the student is ready for feedback, they move to the analysis phase and the system displays both a learner model and some checking notes. This makes VCM a little like a teacher sitting beside the student; ready to report how they interpret the student's map. The interaction with the learner model allows the learner to negotiate that understanding, by checking the elements that VCM suggests for checking and by reflecting on the conclusions captured in the learner model. It also makes VCM very like the many other systems which make the learner model available to the learner as a basis for reflection. For example, it has considerable surface similarities with externalised Bayesian student models [6] and it might be seen as a form of learner model externalisation, for discussion between learners as well as individual reflection [7, 8. 9]. It is also quite close to the work of Dimitrova [10, 11] in that it calls upon the learner to interact with the system to develop and refine their understanding.

2 Overview of Concept Mapping phase

Figure 1 shows the screen of a simple complete concept map. Some features of the concept map that adhere to Novak's pure notion of concept mapping [1] include:

- definition of concepts, which in VCM appear in the panel at the left;
- connection of concepts with links to form propositions such as "circle is a shape";
- definition of links that can participate in propositions;
- the arrangement of the concepts on the map to reflect hierarchy. For example, in Figure 1 "shape" is at the top of the map because it is the most general concept. Similarly, the bottom concepts, indicating the degree sums are below the concepts they relate to;
- use of grouping and horizontal alignment to show that concepts are related. For example, in Figure 1, "circle", "square" and "triangle" are horizontally aligned and near each other because they are all shapes.

Students create a map from a given list of concepts (on the left) and links (on the right). The VCM teacher interface allows the teacher to define an initial set of concepts and link names. Students can add their own concepts or links to the lists if they want to extend their map or incorporate aspects that the teacher did not include.

When they are ready to verify their map, students click on the "Analyse" button in the toolbar and are presented with an output such as that in Figure 2. Since the map in Figure 1 was correct, there were no questions "asked" in this phase. The student can view the user model inferred from the map in the form of the information under "Sentences to save".

The teacher interface for VCM enables the teacher to set up two main aspects of a mapping task. First, it allows them to define the elements of the map: concepts, links and propositions available initially. All of these establish the context of the mapping task. The propositions can be helpful in getting the student started along the general path intended for the mapping task. The second main task of the teacher is to define the actions that VCM will take in the analysis phase runs. The interface which supports this is described in [3]. It has some reasonable default actions but generally allows the teacher to associate an arbitrary feedback string with each element of the feedback. For example, the default question associated with a missing concept is to ask the student if they can see how to include it in the map. For the purposes of student modelling, the critical part of this stage is that the teacher specifies which propositions are to be saved, be these correct or not.

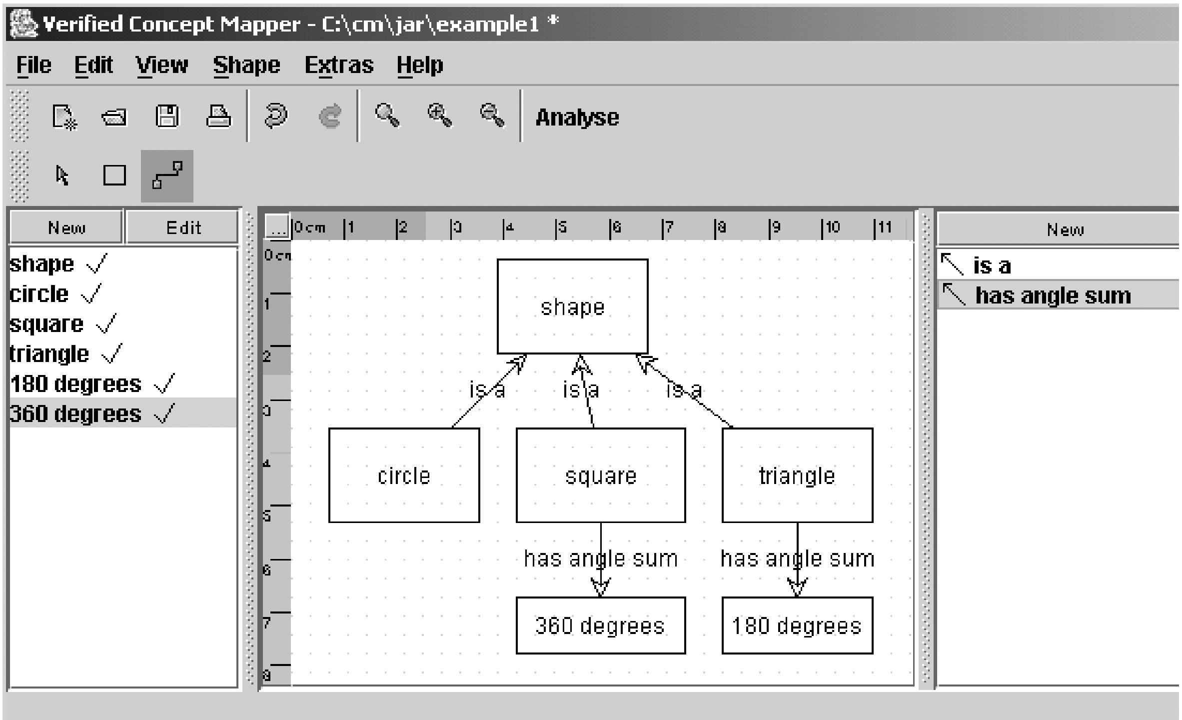

Figure 1: Screen Shot of a Complete Concept Map

In Figure 2, the teacher has specified that "you correctly connected circle and shape" is saved for the proposition "Circle is a shape". Note that the actual text presented to the student is entirely controlled by the teacher. This means that the teacher decides just how the map will be interpreted for student modelling.

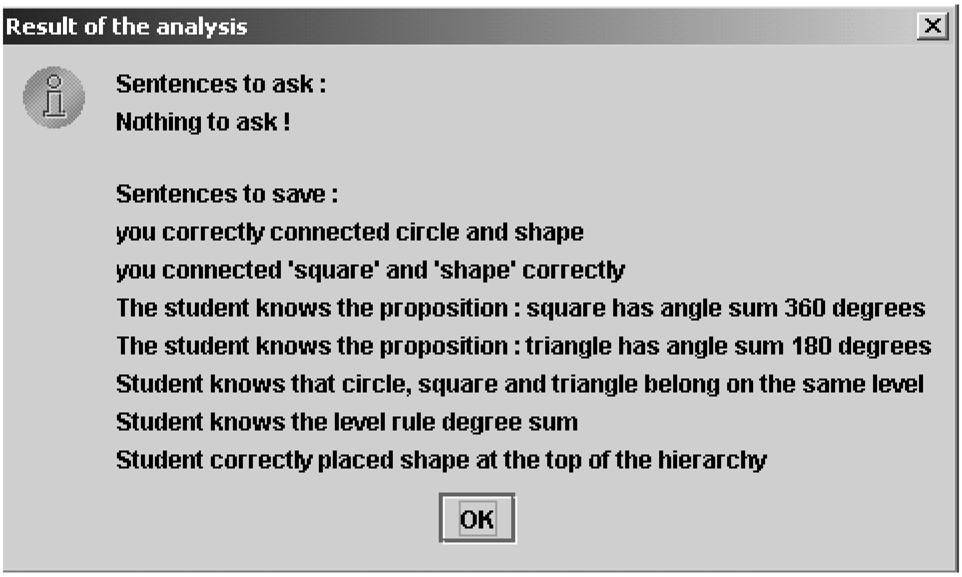

Figure 2: Results of the verification of the map in Figure 1

3 Building a user model from a verified concept map

The software was designed with the expectation that the student would complete the map and then use the analysis to confirm that their map was as they had intended. The "Sentences to Ask" section was designed to highlight any aspect deserving of checking and review. This was particularly intended to help avoid slips based on accidental omissions of a proposition or accidental linkages. The idea was that the analysis would involve a check for the expected propositions. For any that was missing, VCM would produce a message that helped the student check their map. For example, if an expected proposition "Concept1 link1 Concept2" was missing, the teacher might code a message asking the student to consider ways to connect "Concept1". If the teacher anticipated a misconception in the form "ConceptA linkA ConceptB", the message might ask the student to check this proposition. The teacher who created the concept mapping task would decide just how much of this checking they wanted.

The student is able to see how the interpreted map will feed into their student model. This is the "Sentences to Save" section of the analysis report. The teacher may simply choose to report the list of propositions to be saved: they may not choose to signal which are judged as correct, as in our example. So students have access to the information to be saved about them when the use the analysis facility.

At that point, we would hope that they would study this model and check that they were happy with it. Clearly, if the teacher designs the feedback to indicate parts of the model that are considered to be misconceptions, this is an opportunity for the student to check whether they agree with this. Notably, this stage may also offer students an opportunity to check for slips and accidental constructions of wrong propositions. Our design goal was that the analysis output would serve both as a basis for the student to *verify* that their student model was an accurate record of their beliefs and to serve as a basis for *reflection*.

4 Experiments in reflective use of VCM

Our evaluation has been qualitative, based on a thinkaloud [12]. We asked one experienced tutor to perform the experiment so that we could gain input from one person at expert level, but independent of the design team. We also asked four students of first year computer science to tackle the task that is based on their course content.

The mapping task involved scalability, a topic that is quite conceptual and hence suited to concept mapping. It is doubly appropriate because it appears to be quite challenging for students at this level. Participants were asked to begin the experiment by reading relevant materials from the textbook. We provided these as a separate handout. They were asked to follow this reading by drawing a concept map, using VCM. As part of the thinkaloud protocol, they were asked to explain what they were doing as they worked through the task. We observed them throughtout, noting their activity. If they appeared to be concentrating for a long period without explaining what they were doing, we would ask them to think aloud and to state anything that was causing problems. At the end of the task, they competed a questionnaire.

The scalability map was designed by a tutor of the course (one of the authors of this paper) in collaboration with a lecturer who had designed the course and had a particular interest in the problems students have in learning this material. It dealt with:

- scalability at runtime as a function of data size;
- asymptotic worst case cost;
- calculating scalability for nested control structures;
- big Oh notation representing asymptotic worst case cost;
- O(1) < O(log n) < O(n) < O(n logn) < O(n2);
- Actual meaning of each O above.

The design of the mapping task involved two important and quite demanding steps for the teacher/designer. First, we defined the core elements that would be made available to the learners. Then we wrote the analysis rules. The designers of this mapping task were able to draw upon their considerable experience in teaching this topic, and so could define rules which would help identify common misconceptions. They also created several rules designed to help students refine their thinking and to indicate this by improving their maps.

Accordingly, the prompts went beyond the defaults. For example, rather than asking the default question "Check the connections for the concept "O(1)" ", we asked "What big Oh cost grows next most slowly to O(1)?". This question can help the student show the simple increasing runtime of the five Oh costs in this task. Such questions can help the student refine their understanding. This is reflected in simpler, more elegant maps which indicate a clearer understanding. In this case, it allws the map to exploit the transitivity of these Oh runtime costs relationships and so it avoids unnecessary links. For example, it helps the learner see that can just connect O(1) with O(n), rather than trying to connect each cost with every other.

5 Results

The students spent between 1 and 2 hours on the task, while the expert took only 30 minutes. One student failed to complete the task and found it a frustrating experience. None of the students (or the expert) appeared to spend much time reading the supplied reading material. Nor did they make reference to it as they attempted to construct the map. Although we did not explicitly ask the reason for this, it seems that the students may have felt the lecture material was sufficient. The expert knew the material well and had learnt from it as a student.

Figure 3 shows a screenshot of a partially completed map by one of the students. We can see that this map has considerable merit. For example, the student shows understanding of the proposition that O(1) scales better than O(n). However, a better map would put O(1) scales better than O(log n) which, in turn, scales better than O(n). This example illustrates the potential value of the verification phase. This student has shown many of the Oh relationships but they have missed some and have failed to show the transitivity that makes the whole set of relationships clearer and simpler the remember. The verification phase helps the student think about the missing ones.

Note that this example of the relationships between the Oh's indicates how cognitively demanding concept mapping can be. The map in Figure 3 is reasonable but can be improved. As it stands, it is suggestive of somewhat incomplete understanding. If we were to simply infer a student model from this map as it stands, we might well underestimate the learner's actual ability to reason about this subject. The map is reasonably complex, even though it has a quite modest number of concepts and captures the learner's conceptualisation of one, quite coherent and small body of knowledge. We can see that it would be quite easy for slips to go undetected. We can also see that a learner may have difficulty seeing how to make the map more elegant at this point, even if they did suspect that this was possible. The teacher's prompts, both in the questions asked and the student model elements listed is an important basis both for thinking more about that conceptualisation and improving the map.

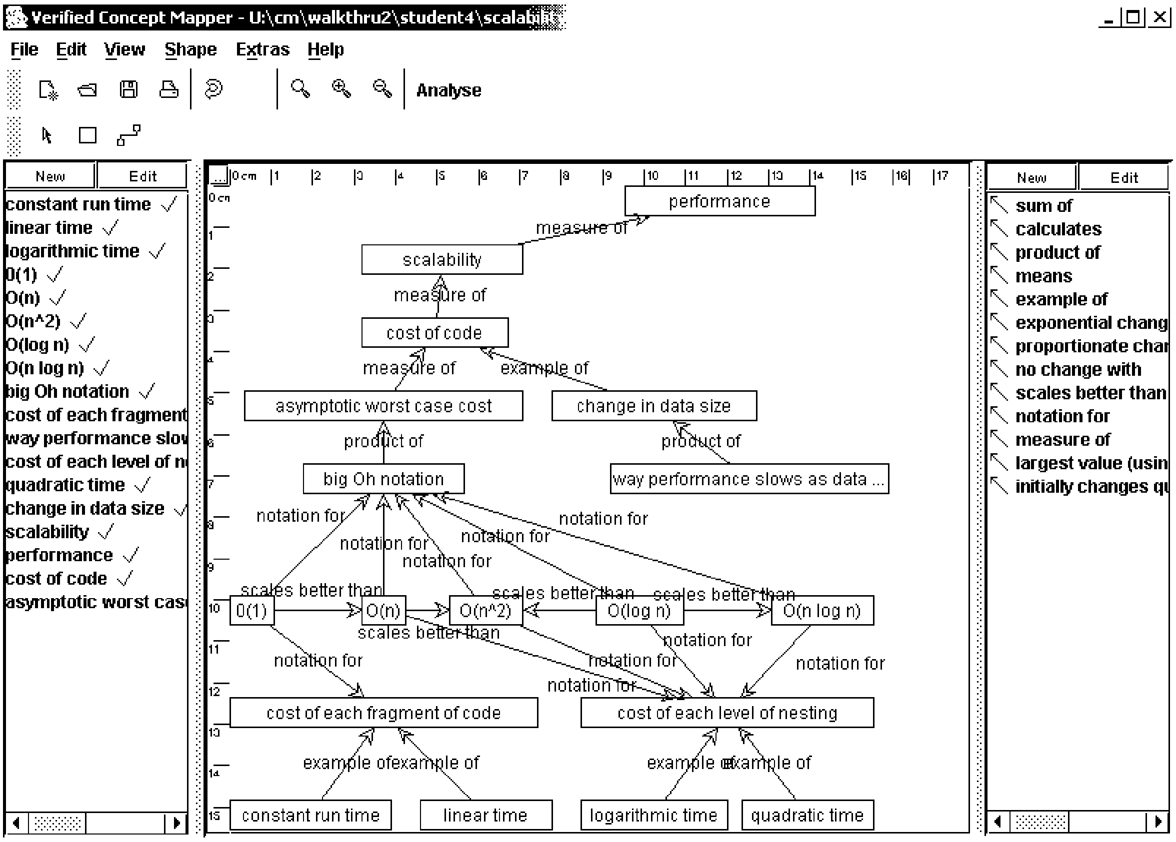

Figure 3: A partially completed student map

Figure 4 shows a screenshot of the corresponding verification output from the analysis phase. The first section, with the `Sentences to ask', is helping the learner see how their map has been interpreted in light of the student model that the teacher intends to extract. In this case, it is a rather long list.

One striking observation was that students used the analysis phase somewhat differently from what we had intended on our design. Rather than wait till they had completed the map and then do the analysis, they used this facility at regular points through the mapping activity. They would do a part of the map, then stop and run the analysis to get feedback on that.

For beginners, this is actually a much better use of the analysis facility that our initial design intent. Concept mapping is cognitively demanding. So, it makes real sense for a learner to do a part that constitutes a logical unit to think about at once. Having completed that part, it makes sense to review, reflect and possibly amend that before proceeding. This then makes it easier for the student to move onto the next part and focus on it, knowing that they can ignore the part that they have now completed. The students' approach also happens to work very well for supporting the effectiveness of the feedback built into the system by the teacher who constructed the concept mapping task and the associated feedback. If students happen to be completely on the wrong track, this approach of getting feedback early on will improve the chances that the student can rethink their map while they have only spent a little time on it.

We were concerned that students might be overly directed by the incremental analysis approach. This is still a concern. However, even in our small trial, we found that when students started out badly, with incorrect propositions, they tended to continue with further incorrect propositions until the map was grossly incorrect.

Result of the analysis

Sentences to ask :
What mathematical relationship does the cost of each fragment have to the overall
What mathematical relationship does the cost of each level of nesting have to the c
What is O(n ^2) an example of?
What is O(n) an example of?
What is O(1) an example of?
What is O(log n) an example of?
What big Oh cost grows the next most slowly to O(1)?
What big Oh cost grows the next most slowly to O(logn)?
What big Oh cost grows the next most slowly to O(n)?
What big Oh cost grows the next most slowly to O(n logn)?
What kind of run time does O(1) represent?
What kind of run time does O(log n) represent?
What kind of run time does O(n) represent?
What kind of run time does O(n squared) represent?
How does constant run time change with a change in data size?
How does linear run time change with a change in data size?
How does quadratic run time change with a change in data size?
What is the definition of scalability?
What is measured by asymptotic worst case cost?
What else is asymptotic worst case cost a measure of?
How does the run time of logarithmic code change with a change in data size?
How does asymptotic worst case cost relate to the total cost of code?
Thinking of the algorithm for finding the cost of each fragment as the product of the
Thinking of the algorithm for finding the total cost by adding the sum of each fragme
Where should the concept "scalability" be in the hierarchy?

Sentences to save :
Student knows the level rule run time
Student knows the level rule big Oh notations

Figure 4: Verification Results for the map in Fig. 3

This incremental analysis approach also means that the student only needs to study a small part of the newly inferred user model information at each stage. So each checking phase is fairly quick. This avoids the possibility of a long list of things to check and user model elements to study if the whole map is analysed in one go.

We observed that the students who used this regular verification approach completed the task more easily and found it more enjoyable. Once we realised this, the think-aloud observer suggested this approach if their early map was seriously flawed. This was intended to reduce frustration.

Each student responded differently to the analysis output. All approaches led to the student focusing on a particular aspect of the task, be it to create a single proposition or to find a pattern to complete a group of propositions. Different approaches included:

- scanning the list of questions for one they could answer;

- scanning for questions for similar groupings. For example, questions 3-6 in Figure 4, all prompt the student to connect "O(1) is an example of big Oh notation" and so on;
- starting with the first question, trying to work it out and then working systematically through all questions they could answer.

Sometimes, the student found a question, which instantly clarified a concept. Generally, they had to think about the answers to each of the questions. It was quite clear that the output of the analysis phase helped the student reflect. Students appeared to find the thinking aloud quite natural. Often, a students would repeat the question to themselves while returning to the map, joining concepts and links out loud before deciding on a proposition. Non-verbal cues that they were doing this included moving the mouse around to rest on concepts/links. One student would even test out a few possibilities by physically linking the concepts, reflecting on the new map form and then, correcting if necessary.

Students also used the strategy of checking what was displayed under "Sentences to Save" to see if they had achieved a correct proposition or arrangement of the map. In this way, they used the information being saved into their user model to reinforce what concepts were correct and use this in their reflection on how to build up other propositions.

The expert user completed the bulk of the map correctly without using the verification analysis. They used the analysis only at the end. At that point, it prompted them to clarify two propositions. So, even for the expert, this phase was helpful in improving the map and the inferences we could build from it. It seemed, from observing this expert user, that our initial design worked well for them. It may be that a very small concept mapping task might be managed similarly by beginners.

6 Discussion

The verification interface was extremely important in supporting user reflection in the students. It stimulated the students to go back to their map and consider what was correct or incorrect about it. While some propositions or arrangements were fixed without much apparent reflection, the majority of them led to the student spending some time thinking about how the concepts linked together.

The "Sentences to Save" portion, which shows the students what information in their current map is being saved to their user model, gave the students positive feedback on what they had correctly stated on their map. Students used this in conjunction with the questions to work out how to create other propositions on the map. In this sense, the explicit display of the user model inferred from the map serves as an aid to reflection.

We are currently planning to collect aggregate statistics on the maps of students so that we can share this with the teacher. This should be as an aid for the teacher's reflection on their teaching. Put differently, if the teacher perceives teaching as an act of learning how to help students learn, the class model will serve as a basis for the teacher to reflect on their effectiveness. The other important need for this teacher-reflection relates to improving the quality of the concept mapping task and the design of the information presented to students and inferred for the student model.

While it is not feasible for the teacher to anticipate and question every possible student error, the teacher could also examine all the propositions made in the map and determine which differed from their own map. While many of these would be valid (eg representing that "O(1) scales better than O(n)", O(n2), O(log n) etc is certainly true.

Several of this student's propositions are invalid. For example, they incorrectly proposed that "O(1) is a product of the cost of each fragment of code". This was not anticipated by the expert and hence is not questioned. In its current form, VCM does not incorporate all propositions into the final user model. The only propositions saved are the ones that the concept mapping task was designed to use to infer an element of the student model.

The main outcome of the evaluative experiment is that the use of incremental analysis should be the main mode of use. This enables the student to stage the review of the learner model and the checking of hints from the teacher. In future work, we will explore the implications for this approach, both on the design of the software and the teacher's task in designing a concept mapping task for VCM. Our evaluations to this stage have been quite limited in terms of numbers but their qualitative nature has provided rich information about the way that VCM can be used. It appears that the availability of the learner model and checking hints provide additional support for reflection in the highly reflective task of concept mapping.

Acknowledgements. We are grateful to Professor Alan Fekete for considerable assistance he provided in defining the map and the mapping task, as well as helping to refine it. We are also grateful to the students who volunteered to participate in the evaluation of VCM, hopefully consolidating their own knowledge of this material while assisting us in improving this particular learning tool. This work was supported the Commonwealth Department of Education, Science and Training Webworkforce Grant.

References

1. Novak, J.D. and D.B. Gowin, *Learning how to learn*. 1984: Cambridge University Press.
2. McClure, J., B. Sonak, and H.K. Suen, *A concept map assessment of classroom learning: Reliability, validity and logistical practicality.* Journal for Research in Science Teaching, 1999. **36**(4): p. 475-492.
3. Gaines, B. and M. Shaw. *Collaboration through concept maps*. in *CSCL95: Computer Support for Collaborative Learning*. 1995. Mahwah, New Jersey: Lawrence Erlbaum.
4. Novak, J.D., *Metalearning and Metaknowledge Strategies to Help Students Learn How to Learn*, in *Cognitive Structure and Conceptual Change*. 1985, Academic Press. p. 189-209.
5. Novak, J.D., *Learning Theory Applied to the Biology Classroom.* The American Biology Teacher, 1980. **42**(5): p. 280-285.
6. Zapata-Rivera, J.-D. and J. Greer, *Inspecting and Visualizing Distributed Bayesian Student Models*, in *Intelligent Tutoring Systems*, G. Gauthier, C. Frasson, and K. VanLehn, Editors. 2000. p. 544-553.
7. Bull, S. and P. Brna. *What does Susan know that Paul doesn't? (and vice-versa): contributing to each other's student model*. in *International Conference on Artificial Intelligence in Education*. 1997: IOS Press.
8. Bull, S., P. Brna, and H. Pain, *Mr Collins: a collaboratively constructed, inspectable student model for intelligent computer assisted language learning.* Instructional Science, 1995. **23**: p. 65-87.
9. Bull, S., P. Brna, and H. Pain, *Extending the scope of the student model.* User Modeling User-Adapted Interaction, 1995. **5**(1): p. 44-65.
10. Dimitrova, V., J.A. Self, and P. Brna. *The interactive maintenance of open learner models*. in *AIED99*. 1999. Amsterdam.
11. Dimitrova, V., J.A. Self, and P. Brna. *Involving the Learner in Diagnosis – Potentials and Problems*. in *Web Information Technologies : Research, Education and Commerce*. 2000. Montpellier, France.
12. Nielsen, J., *Estimating the number of subjects needed for a thinking aloud test.* Int. Journal of Human-Computer Studies, 1994. **41**(1-6): p. 385-397.