Concept mapping for eliciting verified personal ontologies

Laurent Cimolino, Judy Kay* and Amanda Miller

Smart Internet Technology Research Group,
School of Information Technologies,
Madsen F09, University of Sydney, Australia 2006
Fax: +61-2-9351-3838   E-mail: judy@it.usyd.edu.au
*Corresponding author

Abstract: Concept mapping is a valuable technique for education evaluation. This paper describes VCM, a system based on this approach: it supports teachers in creating concept-mapping tasks intended to capture the student’s understanding or personal ontology of a small domain. A novel feature of our work is that the system verifies whether the student intended the map elements that will be used to infer their understanding and misconceptions. We provide a detailed description of VCM, linking it to related work and describe its qualitative evaluation.

Keywords: concept mapping; personal ontologies; knowledge elicitation; student modelling; self-explanation.


Biographical notes:
Laurent Cimolino has an Engineering degree in Computer Science and Applied Mathematics from the Ecole Nationale Superieure d’Electrotechnique, d’Electronique, d’Hydraulique, d’Informatique et des Telecomunications (ENSEEIHT), Toulouse, France. He worked on this project as an intern with the Smart Internet Technology Research Group at the University of Sydney, Australia. He is currently working at Amadeus, France.

Judy Kay is an Associate Professor at the School of Information Technologies at the University of Sydney, Australia. She is a Principal in the Smart Internet Technology Research Group and leads projects in intelligent teaching systems and customising and filtering of information. She has over 100 publications in the areas of personalisation and teaching and learning.

Amanda Miller is currently a student in the Graduate Medical Programme at the University of Sydney, Australia. She has completed a BSc (Bioinformatics) degree. She has worked as a Research Assistant on a range of projects associated with innovation in support for learning, within the Smart Internet Technology Research Group at the University of Sydney.

Copyright © 2004 Inderscience Enterprises Ltd.
1 Introduction

Concept maps [1–4] have become a common tool for externalising learner conceptions of a domain. In doing this, learners reflect on their understanding of the domain and gain access to their own, often unconscious, conceptualisations. There are many available tools for learners to draw concept maps, such as those summarised at [5,6]. Concept mapping has strong foundations in theories of learning and in empirical studies of brain activity [4]. The approach has many potential roles in education.

We are concerned with the use of concept maps as a mechanism for determining the way that a learner conceptualises a domain. From this, we aim at building accurate and detailed models of the learner’s conceptual knowledge of a domain. Since our research focuses on scrutable learner modelling and strong learner control, the concept map is a natural tool for eliciting learner models.

In line with our philosophy of learner control, our current work explores approaches to building verified concept maps for the purpose of modelling the student’s knowledge. We consider it critical to verify concept maps before using them as a basis for reasoning about the student’s knowledge. This is because it is very easy for a student to accidentally link the wrong concept or omit a concept or a link from their concept map. Moreover, Novak and Gowin [2] emphasise that revision of concept maps is a normal and important part of the concept-mapping process: they comment “it is always necessary to revise [the original map] … Good maps usually undergo three to many revisions”.

This paper describes VCM, our system for verified concept mapping. Section 2 describes the interface from a student’s point of view, Section 3 from a teacher’s. Section 4 reports our qualitative evaluation of VCM. Section 5 discusses the findings of the evaluation experiment and describes how VCM is related to other research. Section 6 concludes the study.

2 The student interface

Figure 1 shows an example of the interface as a student might find it at the beginning of a concept-mapping task. In this case, the teacher has set up the task with a partial map to get the student started. We now give an overview of the interface.

The top tool bar shows the commands. The File menu has the usual commands to save, open, export and print concept maps. Edit allows the user to undo and redo actions. View manages the display of the graph: the user can view the graph with or without the grid, the rulers or the page layout. There is also a zoom. Shape supports organisation of the graph elements: aligning them in any direction as well as grouping and ungrouping. Extras sets the value of the grid and gives an overview of the graph in a pop-up frame.
The second tool bar supports short cuts. It also has the Analyse command, which the student selects on completing a stage in a concept-mapping task. This starts the process of checking the map after which the system asks questions which are intended to verify the map drawn. We will return to this aspect: it is core to the verification of the concept-mapping process.

In the next row are icons for drawing the concept map. The button labelled with the arrow symbol \( \uparrow \) is the default action. It allows the user to select, unselect and move elements of the map. The \( \downarrow \) is used to add a new concept in the map and \( \leftrightarrow \) is used to add a new link between two concepts. Normally, the teacher will have created the concepts and links and these will be visible to the student in the left and right panels, respectively. However, if the student really feels the need for additional ones, the interface does allow students to create new concepts and links so that they draw the concept map as they feel it should appear. As we discuss below, the system can only verify the use of concepts and links that the teacher had anticipated and made available to the student.

The very bottom bar of the interface is a status area. It provides an explanation of the element the mouse is pointing to. Figure 2 shows an example of a concept map that might have been created by a student. We have used colour to indicate errors and will refer to these later.
Concept mapping for eliciting verified personal ontologies

Figure 2  Example of a map that has been constructed by a student

Normally, the student would create the map in two stages. First, they would place the concepts on the map. This involves selecting them from the menu at the left and placing them. Then, the student would typically go through a stage of moving the concepts around on the map, getting the layout to really reflect the structure desired. The concept-mapping process requires that the student should place higher level concepts higher on the map and more specialised ones lower. Similar level concepts should be at the same level. Also, concepts that are thought of as similar should be placed together. It is also desirable to ensure that the layout is pleasing. As this description suggests, this layout process is quite cognitively demanding and will typically require many revisions and refinements. Essentially, this stage of the process involves externalising part of the student’s ontological understanding of the domain as it is supposed to capture the hierarchical relationships between concepts, similarity relationships and symmetry relationships. It also involves laying out the concepts so that it will be easier to connect the concepts that will be linked. So, concepts that are related will need to be placed in reasonable proximity to each other.

In the second stage, the student completes the rest of the ontology externalisation. This is the construction of the propositions. For example, the initial map in Figure 1 has the proposition *Plants have roots*. Propositions constructed with this interface have two concepts connected by a link. During this second stage, it will be common for the student to move the concepts around to reduce crossing of link lines and to improve the appearance of the map.

When the student thinks their map is complete, they click the *Analyse* button. First the program checks if each concept has been included in the map. If not, the analysis stops and the program shows a list of missing concepts.
Once all the required concepts have been used, the Analyse phase checks other aspects of the map. First and most important are the propositions that are the essence of the map. For example, the top part of the map in Figure 2 represents the proposition *Plants have Roots*. If the student’s map is missing a proposition that the teacher expected in the map, this analysis phase will result in a message, which hints about the missing proposition. This type of message is illustrated at the top of Figure 3. The actual details of these messages are entirely controlled by the teacher who sets up the concept-mapping task. It is quite feasible to allow for different understandings and associated propositions. It is also the teacher’s decision to have more or less checking of the map and corresponding numbers of messages.

**Figure 3** Example of messages produced at analysis phase for the concept map of Figure 2

---

Figure 3 also provides examples of system messages, which indicate the inferences the system draws from the concept map. This is the set of statements listed under the heading *Sentences to save*. The first block of these is the propositions that the system has recognised and which the teacher has specified for saving. For example, the first indicates that the result of the mapping activity will include a record that the student knows the concept *Plants have Stems*. The fourth last statement, *Student knows the level rule Flower components*, is an example of an inference about the way that the student has laid concepts out at the same level to indicate that they are similar. The last three statements all indicate that the layout of the map has been used to infer that the student understands the hierarchical relationship between the sets of concepts. The reason we have included this screen is so that the student can see the complete set of inferences that have been made about their map.
3 The teacher interface

The teacher uses an interface like that shown in Figure 4. It is very similar to the interface that the student will see. However, there is an additional Teacher menu. This supports special functions needed to create a new concept-mapping task. These include definition of the parts of the teacher’s map to be made available to the student at the beginning of their mapping activity. It also provides access to the parts of the interface needed for defining the analysis process for a map. We now describe the process a teacher would go through in order to create the map shown in Figure 4.

Figure 4 Example of a mapping task being developed by a teacher

3.1 Creating concepts and links

The first stage in creating a concept-mapping task is to define the concepts that will appear at the left of the screen and the links that will appear at the right. Selecting the New button above either list brings up a screen for creating a new concept. Creation of links requires the teacher to define whether the link is unidirectional as in the case of links such as has or bi-directional as in the link, synonymous. The interface also requires the teacher to provide additional information for use in subsequent student modelling: the concept or link must have a creator and an explanation. The latter is potentially useful for capturing some of the subtlety of the teacher’s reasoning in the choice of a particular term. For example, it may be that one term was used in the textbook or another was a common source of difficulty in the past. Perhaps there is an unusual term that the teacher prefers.
Essentially, this is the stage at which the teacher is defining the vocabulary underlying their ontological understanding of the area covered by the concept-mapping task. Any reasoning that was important in defining the vocabulary can be captured at this stage. The Edit menu allows the teacher to later edit these concept and link definitions as needed.

3.2 Building up the mapping task

The teacher can perform the bulk of this process in the same way as a student drawing a concept map. We wanted this to be the case so that the teacher would be able to have the same look and feel as the student would have when doing the task.

The teacher can also create propositions using the New Proposition button in the tools bar. This is essential for the definition of propositions that constitute alternate conceptions or ontological views of the subject. The teacher needs to define these so that the analysis phase can deal with incorrect or other alternate propositions that the teacher expects students to create. This aspect is important in enabling the verified concept-mapping system to elicit student models for such misconceptions or alternate views. Figure 5 shows the screen that can be used to define new propositions. This provides the complete list of current concepts at the leftmost and rightmost panels. In the middle is the complete list of links. Selecting one item from each of these enables the teacher to define any proposition that is possible for the current set of concepts and links.

Figure 5 Creating arbitrary propositions and adding analysis actions to them

3.3 Creating analysis actions

Each link on the visible map can be used to select a proposition. Typically, the teacher would go through each such proposition in their own map and the additional propositions they have defined and, for each, define the actions that should occur in the analysis phase. The Teacher menu gives access to a complete list of the propositions. This provides access to the interface screen for defining these actions as illustrated in Figure 5.
Each new proposition has two default processing actions. These are illustrated at the bottom of Figure 5, labelled with the number 1. The first indicates that if a student correctly creates this proposition, the output of the concept-mapping task will result in saving the proposition as known. The second is to automatically ask the student to consider amending their map if the proposition is missing from their map. The default text for this is shown in the figure. As the interface has a text box for these messages, the teacher can choose to alter them to any arbitrary text of their own choosing.

An example of the screen for defining a misconception is illustrated in Figure 6. The interface supports creation of analysis rules for the vertical layout of the concept map. These are created in much the same way as the processing rules for propositions. There are two types of rules, one for checking that a set of concepts have been placed level with each other, and the second checking for a hierarchical placement with one set of concepts above another set. A level rule has only a name, a set of concepts and its processing actions. The teacher needs to select the level rule button in the tool bar. Figure 7 shows a set of level rules. For example, it indicates there is a rule called Plant components, marked 1 in the figure, and it has the three concepts, Roots, Stems and Leaves, marked 2 in the figure.

Figure 6 Creating a proposition to reflect a misconception. Note that the teacher who has created this screen has not chosen to indicate that they consider this to be a misconception. This may be because they do not want to flag their interpretation as it might influence the learner unduly. Another teacher might choose to suggest that the student check that they really intended this.
To create a new level rule, the teacher selects the New button in Figure 7 and is presented with the window in Figure 8. This operates similar to the analysis rules for propositions. The first default rule saves an indication that the student correctly built a map with this level rule satisfied when this is the case. Where just one or two concepts are missing, the default rule asks the user to check the map, with the teacher providing the wording of the question. The teacher can alter the message and processing action. The buttons at the bottom of the window also allow the teacher to create additional rules. For example, there might be a Save action as well as an Ask.

Figure 8 Window for defining the processing actions for a level rule
The third, and last, rule type checks for hierarchical structures in the map. A hierarchy rule has a name and two set of concepts, those concept(s) above and the ones below. Figure 9 shows the interface for creating such rules. Initially, the set of concepts all appear in the panel at the right. The teacher selects those for the appropriate windows at the left, so specifying one set of concepts, which should appear above the set specified as belonging below. The processing operates similar to the cases already described.

Figure 9  Window for creating processing actions for a set of hierarchical relationships

In the processing, the program determines the relative positions as follows. It finds the smallest rectangle containing all the selected concepts. Then it attempts to separate that rectangle into two by cutting horizontally so that every concept in the top rectangle is in the above-set and all concepts in the bottom rectangle belong to the below-set. If it succeeds in finding such a split, it judges the student’s map to pass on this rule. This is quite a generous interpretation which allows the student considerable flexibility in the actual layout.

Note that this means that the system supports three classes of ontological relationships to be represented and interpreted by VCM. The propositions capture named relationships between a pair of concepts. The level rules indicate similarity or grouping of concepts, generally in a semantic sense. The hierarchical rules will generally capture generalisation/specialisation or whole/part relationships without this being explicitly stated.
The system has been implemented in Java and requires at least Java 1.4. It is based upon the JGraphpad (http://www.jgraphpad.com) open source project and uses JGraph (http://www.jgraph.com) Swing.

4 Qualitative evaluation

We chose to evaluate the system using a thinkaloud, qualitative evaluation, with the recommended [7] three to five participants needed to gain a sense of how users are able to work with the system. Four participants were students studying a summer stream of the relevant first year computer science course and one participant was an expert user who had previously tutored the course. We chose to include an expert user in order to see how a user with an established personal ontology would use the software, compared to student users who were just learning about this area.

The subject of the activity was scalability of software, an assessable topic in our introductory computer science course at first year level. We chose this area for two reasons: firstly, students often find it difficult and demonstrate a poor understanding of scalability. Secondly, it fits well into the propositional form used in concept mapping.

Four students and one expert participated. The students were asked to first read appropriate pages in the recommended textbook and then practice use of VCM interface with a tutorial example. Then, they were asked to complete the experimental task of constructing the scalability map. During this mapping, they were asked to think aloud, explaining what they were doing and reporting any difficulties. At the end of the task, each participant filled out an anonymous questionnaire.

The scalability map was designed by a tutor of the course (one of the authors of this paper) and verified by a course lecturer. It covered the following concepts in scalability:

- definition of scalability as the way performance slows as data size increases
- asymptotic worst case cost as a measure of scalability
- the asymptotic worst case cost totalling the sum of the cost of fragments of code multiplied by the cost of each level of nesting
- big Oh notation representing asymptotic worst case cost
- \( O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) \)
- \( O(1) \) is notation for constant run time, \( O(n) \) for linear run time etc.
- Change in performance with change in data size (e.g. code with constant worst case cost has no run time change with a change in data size).

The map designer specified detailed questions above the most basic level of the course. These were designed to provide an additional layer of support to the student. The majority of questions were open-ended and designed to help the student see a concept differently. The remaining questions were designed to guide the student so that they would avoid unnecessarily clogging up the map. For example, the question, “What big Oh cost grows next most slowly to \( O(1) \)?” helped students avoid representing that \( O(1) \) was less than each of the other big Oh costs.
4.1 Results

Four of the five subjects finished the task. The expert user took about 30 minutes but the other students took between 1 and 2 hours. The student who did not finish the task (Student 3) seemed very frustrated with all aspects of the activity and did not seem open to adapting to a different way of thinking.

Figure 10 shows the final map drawn by the expert. It mirrors that of Students 1 and 2. Figure 11 shows the final map drawn by Student 4. This map contains all the correct propositional links and was thus verified as correct by the system. However, this map also contains a few misconceptions that were not anticipated by the teacher who created the mapping task, and hence, were not caught by the system.

**Figure 10** Completed expert map

**Figure 11** Student 4’s completed map
Note that VCM does not direct the student towards building an expert map, such as that in Figure 10. What it does do is to provide feedback so that a learner can verify that their map is what they intended. The two main forms of output from the analysis phase serve to support this verification in different ways. The first block of output, the Sentences to ask, indicates aspects the teacher had expected in the map. This should help the student to check these elements of the map. The second set is a summary of the elements of the map that the teacher has chosen to save. These serve as another level of checking since they are the aspects the teacher anticipated and considered most critical in the map. The verification phase invites the student to check that they actually intended these. We believe that this verification is particularly important and valuable in concept mapping because it is so cognitively demanding. It is easy for the student to make slips. In a map such as those in Figures 10 and 11, there is also sufficient detail that learners may not notice unintended features of the map. If they create a poor and messy map, they may not be even able to see the main elements. In this case, a set of well-designed checking questions from the teacher may help the student see the domain more clearly and so to produce a better map.

4.2 Questionnaire results

The students were then asked to rate general aspects of VCM. Table 1 indicates responses to the questions: How satisfied are you with VCM overall? (1 = not at all, 5 = extremely); and, how easy did you find VCM to use? (1 = very difficult, 5 = very easy)

<table>
<thead>
<tr>
<th></th>
<th>Overall satisfaction with VCM</th>
<th>Ease of use of VCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student 1</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Student 2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Student 3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Student 4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Expert 1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Student 1 commented during the walkthrough that they really relished using online tools and tutorials, which may explain their high level of satisfaction with the software. When asked if they felt that the tool allowed them to more completely represent their knowledge than if they were taking traditional notes, one student said that it did, one said it made no difference and the other said no. Only Student 3 felt that the software did not adequately test their knowledge of the subject, saying

“Although it may have demonstrated some discrepancies in my knowledge, the manner in which propositions were made often confused rather than simplify the process.”

By contrast, other students commented

“the links helped to clarify the relationships and meaning of words & concepts”,

“I can write down the main points in the map and when it comes to revising, I can test myself by looking at those keywords and see if I know what those words are referring to,”
“it does give me a better idea of the whole picture of scalability”

“it covers the most important parts of scalability and the relationships.”

Perhaps the most interesting observation related to the way that the students used the analysis. The more confident users did not want to use the analysis until a substantial portion of the map had been completed. While this tactic was appropriate for the expert, it was disastrous for the students who spent a great deal of time creating a poor map. The observer prompted the user to check their work if their map was becoming grossly incorrect. At the first prompt, the students looked at the map and said it was right. A few minutes later, the observer asked them how they would know if their map was right and at this point they usually checked using the Analyse button. When asked why they waited so long to analyse their map, the students commented

“I like to get a whole idea of the map before checking”

“it’s like compiling – I need to get big chunks done before I want to compile.”

This is consistent with our observations of many students as they program and it is a dangerous practice, which often costs them considerable wasted time. Once these students saw the enormous list of questions, they then began frequently consulting the analysis, indicating that they either realised their strategy was incorrect or were not entirely committed to it! Student 3 commented that they did not want to analyse early because it would slow them down. In fact, this strategy slowed them down immensely because they constructed a seriously flawed map; this would have taken a long time to correct had they not given up. Once the other students were aware of the analysis, they used it frequently, either checking between every new proposition created or creating multiple like/related propositions and then checking. The expert was able to proceed with the mapping task without the benefit of feedback from the analysis phase; this is the way that the students would have liked to work but their lower level of knowledge made that inappropriate.

5 Discussion

Our system allows the student to construct a map, and then, once they decide that a stage of the mapping is complete, they can indicate this at the interface. The system then checks the map and may prompt the student to review selected elements. This helps the student verify that they really are content with the final form of the map. This phase of the mapping activity can help the learner check that they have not made careless mistakes. It also helps them think about additional links and conceptual relationships that they actually know about but may have failed to include in the initial map. Since concept mapping is such a cognitively demanding activity, it is highly likely that the learner will forget some of the map elements that they actually do know about. This prompting phase helps them to return to the task and focus on elements they have forgotten. At the same time, the output of the analysis could well push the student to consider aspects of the teacher’s ontology. Less confident students may use the checking phase of the map for help in finishing the map to the level intended by the teacher.

Effective use of the approach draws upon the expertise of the teacher, both in their knowledge of the domain and their knowledge of common alternate understandings and misconceptions at the propositional level. The closest work is the Giant [8] where the
system collects a soup of propositions from a group of students and uses these as a basis for prompts to reflect. Our approach aims to exploit expert knowledge of the teacher, both of the domain and of important misconceptions such as those that might be repeatedly observed in the classroom and in examinations. It also has some elements in common with [9]: important differences between that proposal and VCM are that the former uses deep qualitative reasoning where we only allow reasoning explicitly provided by the teacher. For our context, where we want to be able to infer a student model, this constraint makes it straightforward to make the mapping activity fit a particular learning context that is well understood by the teacher. Indeed, we treat the maps as Novak described them [2–4] as a means to externalise propositional knowledge in a defined learning context. In that situation, it is dangerous to automatically assume that students intend inferences based upon the usual semantics of links.

It is quite possible that the teacher might construct feedback that learners might disagree with. There are two main possibilities to consider here. First, it may be that the learner and teacher have different conceptualisations. In that case, the teacher’s feedback alerts the student to the fact that there is a difference. What happens next depends upon the teacher, learner and the learning environment. The second possibility is that the teacher and learner actually do have the same conceptualisation but the teacher has accidentally coded incorrect responses to correct mapping elements. For just this type of situation, we are currently planning to capture and analyse all logs of such information. This can then be made available to the teacher who may then identify the problem and fix it.

The VCM project has strong links with the growing body of research into making student models open to the learner [10–13]. One important aspect of such work is to encourage student reflection. VCM does this on two levels. First, concept mapping encourages reflection, as the student externalises their understanding [2–4]. Secondly, the VCM verification stage is somewhat like the negotiation between the student and the system’s view of them in the student model: this negotiation process encourages the student to focus on salient parts of the map and is analogous to these systems that encourage negotiation with an open learner model.

Now consider the other important role of open learner models: they should have the potential to improve the accuracy of the model [10–13]. On the surface, this appears quite similar to VCM. However, there is an important difference. We do not intend the concept map as student model; rather, we see it as just one source of information about the student. In our user modelling framework [12], VCM constitutes just one source of evidence about the learner. Other learning activities, tests and associated information will contribute to the learner model, rather as described in the open Bayesian student models of [13]. Indeed, several propositions from the map may all contribute to a single component of the learner model. For example, in our evaluation experiments, all the propositions could contribute to a single component, representing the learner’s understanding of the overall concept scalability. This is important since the student interacting with VCM is not seeing their student model; they are seeing one source of evidence about it.

Our interface has been designed to remain true to the spirit of the original work on concept mapping by [12]. The student is able to create their own concepts and links if they wish. This means that they are not constrained. At the same time, if the student does keep to the set of concepts provided by the teacher, they may gain feedback on how that teacher had expected them to draw the map and some of the essential elements expected.
Essentially, the interface supports the student in externalising their ontology of the domain of the concept-mapping task. The teacher’s set of defined concepts and links as well as the initial parts of the map should help ensure that the student takes a perspective in line with that the teacher intended. Once the students have completed their own map, the analysis phase enables them to see the differences between their conceptualisation and the ones the teacher expected. The questions that the student is asked help them reflect upon their own ontology and the elements that differ from the teacher’s.

6 Conclusions

We have constructed VCM, a system for eliciting student’s conceptualisation of a domain. Qualitative user trials, conducted as a think-aloud experiment, indicate that the approach is promising. Perhaps most importantly, the evaluations indicate that the analysis of a student’s map allows the teacher to help the student to check for careless errors and that the propositions reflecting misconceptions are actually held. For the purposes of scrutable student modelling, it is important that VCM shows the student just what is inferred from the mapping activity. VCM enables the teacher to provide the student with access to carefully chosen parts of the teacher’s own ontology. It also allows the teachers to exploit their own expert knowledge of common misconceptions and allows them to decide just how they wish to present them to the student. One of the limitations of our approach is that VCM operates largely at the propositional level. This has been a conscious decision, keeping VCM in line with the way Novak conceived concept mapping. VCM also supports the role of concept mapping as a form of knowledge construction because its analysis phase give pointers for the learner to use in reflecting on the map they have drawn. The cognitive demands of concept mapping are heavy. So, one can expect that learners will often make unintended mistakes in their maps. Perhaps they failed to complete part of the map because they were side-tracked on another part of the map and forgot to come back and complete it. Perhaps they accidentally connected concepts or inadvertently selected the wrong link. Perhaps they opted for a poor layout and this meant that they failed to think through as full a map as they might be able to do. With good feedback, from a well designed VCM mapping task, the learner can gain assistance in recognising these problems. Then, the verification phase can help them develop a more complete map, which is a better, more accurate articulation of their conceptualisation.

References

228 L. Cimolino, J. Kay and A. Miller


