Concept Map Assessment for Teaching Computer Programming

Jeroen Keppens
Department of Computer Science
King's College London

David Hay
King's Institute of Learning and Teaching
King's College London

Abstract

A key challenge of effective teaching is assessing and monitoring the extent to which students have assimilated the material they were taught. Concept mapping is a methodology designed to model what students have learned. In effect, it seeks to produce graphical representations (called concept maps) of the concepts that are important to a given domain and how they are related, according to the students. In recent decades, various methods have emerged to evaluate concept maps, each measuring different features of concept maps. New approaches are still being developed. Few guidelines are available regarding the method to choose for a given application. This paper is a literature review that consists of two parts. The first is a review of the many automated and manual techniques of concept map analysis. The second is a critical and reflective commentary on these techniques.

1 Introduction

An important aspect of effective teaching is careful assessment of the extent to which students have assimilated the material they were taught. Novak [Novak2005] has devised a concept mapping methodology for this purpose. More precisely, concept mapping has been developed as a tool to assess student learning in a longitudinal to test the benefit of teaching abstract science concepts at an early age.

Generally speaking, a concept map is a graph containing labelled nodes and labelled arrows between pairs of nodes. The nodes represent concepts that are identified by each node's label. The

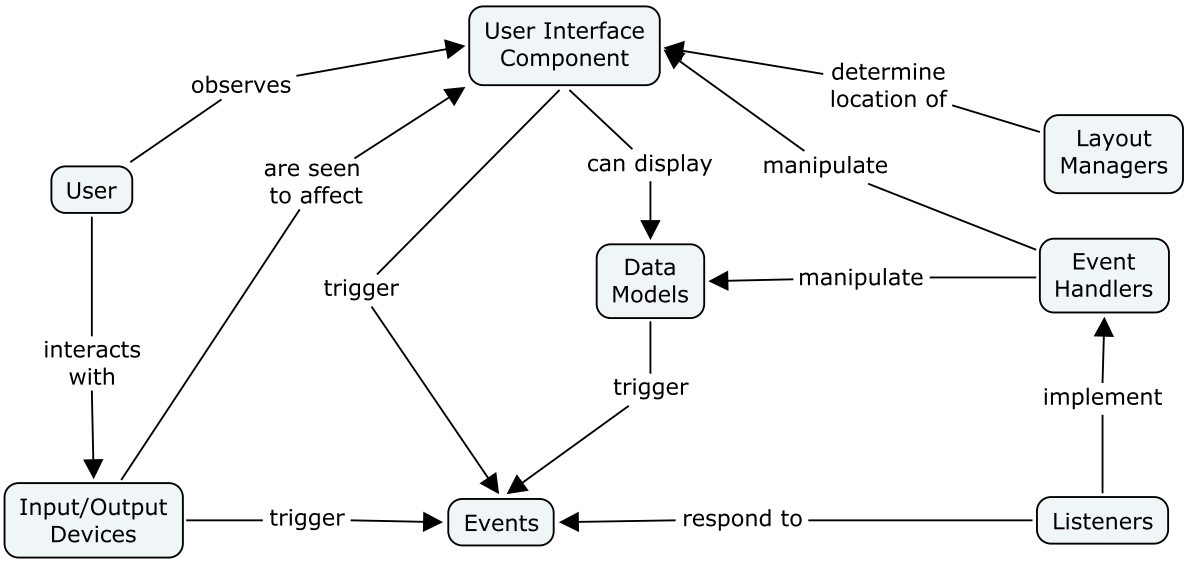


Figure 1: A sample concept map for user interface component

arrows denote relationships, which are identified by each arrow's label, between the concepts they link to one another. Figure 1, for instance, shows a sample concept map describing the notion of a user interface component (in the context of a course on graphical user interface programming). For example, it contains the concepts of "Layout Manager" and "User Interface Component" and a relationship describing that "Layout Managers" determine the locations of "User Interface Components".

The development of this representational formalism was motivated by Ausubel's assimilation theory [Ausubel1968]. Ausubel hypothesised that learning is crucially dependent upon the learner's pre-existing awareness of concepts and their inter-relationships. This implies that any meaningful learning stems from the interaction between newly introduced information and the learner's prior knowledge. He suggests that much of this interaction involves differentiation between alternative meanings and conflict resolution between new and old ideas. As such, the outcomes of meaningful learning may include a more precise classification of concepts and relationships, and the resolution of ambiguous or incorrect ones. Because concept maps are explicit visualisations of the concepts and relations between concepts as seen by individual learner, they are an effective means to test Ausubel's hypothesis.

Of course, little is understood about the way the human brain stores and updates information and knowledge. However, various evidence suggests that it structures information into hierarchies and networks similar to those visualised by concept maps [Bransford et al.1999]. This indicates that students should be able to produce concept maps that are accurate representations of their understanding of a given domain, if the concept mapping task they are set is sufficiently clear.

This hypotheses is corroborated by the work of many researchers, including Novak who has shown that even young primary school children can become proficient at constructing concept maps with relative ease [Novak2005].

The latter does not imply that constructing accurate concept maps is easy. Many studies of concept mapping show that concept mapping is troublesome for many students because it tests personal understanding rather than knowledge that was merely learned by rote [Hay et al.2007, Kinchin et al.2005, Kinchin and Hay2000]. In testing this, however, concept mapping also invites students to engage in the process of meaningful learning and to construct meaning for themselves. The concept mapping task, based on which students are expected to produce concept maps for further evaluation, is another factor that substantially affects the success of the exercise. That is, the question or concept that students are expected to address by means of a concept map, and the tools they are given for that purpose influence what aspects of student understanding are being recorded.

Since their invention, concept maps have gained increasing popularity as a learning and organisational tool. In the domain of computer programming education in particular, research suggests that the development and incremental refinement and improvement of mental models of programming concepts and developed systems promotes meaningful learning [Mayer1981]. However, the potential to assess a student's evolving understanding of the domain of study, in this context, is often ignored [Ruiz-Primo and Shavelson1996]. Moreover, assessment of concept maps remains a topic of ongoing research.

This paper discusses a range of different concept map assessment methods that have been proposed in the literature. It aims to examine their main feature and evaluate their suitability for teaching computer programming. Section 2 is a review of the many automated and manual techniques of concept map analysis. Section 3 is a critical and reflective commentary on these techniques. It is not underpinned by empirical data but it is written to encourage future research, including our own, into concept mapping for teaching computer programming. Section 4 concludes the paper.

2 Assessment methods

2.1 Quantitative assessment methods

Quantitative assessment techniques provide a means to calculate a numerical score for a given concept map as a measurement of a students understanding of a particular domain. The objective of such methods is to produce a total order of the different learners' understanding of the domain or numerical data that can be employed for statistical hypothesis testing. This subsection discusses

Concept map feature	Score
Valid hierarchical link between concepts	5 points each
Valid cross-link between concepts on different branches of a hierarchical structure	10 points each
Other valid links between concepts	2 points each
Examples of concepts	1 point each
Invalid concept or link	0 points each

Table 1: Structural scoring

a representative set of this category of methods.

2.1.1 Holistic scoring method

The holistic method instructs (expert) assessors to assign concept maps a score on a given scale, say 0 to 10, which expresses the concept mapper's overall understanding of the domain. It does not supply any algorithm, heuristics or guidelines to calculate the score. The holistic method was originally devised by McClure, Sonak and Suen [McClure et al.1999] as a control method to test the effectiveness of weighted average methods.

2.1.2 Weighted component scoring methods

Weighted component scoring methods assign partial point scores to certain concepts and/or links between concepts. The score associated with a concept map equals the sum of the partial point scores awarded to each component of that concept map. The values assigned to components may depend on their validity or the type of structure they add to the overall concept map. Two weighted additive component scoring methods have received substantial attention in the literature on concept map scoring: the structural and relational scoring methods.

The *structural scoring method*, devised by Novak and Gowin [Novak and Gowin1984] seeks to reward hierarchically structured knowledge. The method proposes a relatively small score for each valid link and each valid example of a concept. It rewards a substantially higher score to links that express a hierarchical relation, such as "is a kind of" or "contains" relationships. The highest scores are reserved for links between concepts that are located on different branches of a hierarchical structure. Table 1 summarises a sample structural scoring scheme.

The *relational scoring method*, devised by McClure, Sonak and Suen [McClure et al.1999], awards points to each link between concepts in isolation. Higher scores are assigned to links that are correctly labelled and ones that express a foundational relationship of the domain, such as taxonomical and causal relationships. Table 2 summarises a sample relational scoring scheme.

Concept map feature	Score
Valid, but incorrectly labelled link between concepts	1 point each
Valid and correctly labelled link between concepts that does *not* represent a hierarchical, causal or sequential relationship between concepts	2 points each
Valid and correctly labelled link between concepts that does represent a hierarchical, causal or sequential relationship between concepts	3 points each
Link between concepts where no relationship exists	0 points each

Table 2: Relational scoring

2.1.3 The closeness index

The closeness index, devised by Goldsmith, Johnson and Action [Goldsmith et al.1991], is a heuristic that aims to calculate the similarity between a student's and a teacher's concept maps. The approach focusses on the concepts and links between concepts that two maps have in common, but it ignores the labels of the links. The closeness index of a concept c that the student's and teacher's maps have in common equals the number of concepts directly linked to c in *both* maps divided by the number of concepts directly linked to c in *either* map. The overall closeness index of two maps is the average closeness index over all nodes in those maps. In other words, the closeness index equals:

$$\frac{1}{n}\sum_{i}^{n}\frac{S_i \cap T_i}{S_i \cup T_i}$$

where n is the number of concepts that appears in one or both maps, S_i denotes the set of concepts directly linked to the i^{th} of the n concepts in the student's map, and T_i denotes the set of concepts directly linked to the i^{th} of the n concepts in the teacher's map.

2.2 Qualitative assessment methods

Qualitative assessment methods produce descriptive assessments of concept maps. Rather than aggregating concept map features into a single number, they make a synthesis of the various features and provide a descriptive diagnosis of the underlying extent of understanding. This subsection discusses a representative set of such methods.

2.2.1 Linkage analysis

Linkage analysis, devised by Liu, Don and Tsai [Liu et al.2005], aims to identify potential misconceptions of students by comparing the concepts each individual concept is directly linked to in

a student's and the teacher's concept map of a particular domain. In this way, linkage analysis identifies certain symptoms that indicate potential misconceptions and may be able to suggest improvements to flawed concept maps.

For example, linkage analysis can identify potentially confused concepts. If a concept c_1 in the student's map is linked to a set of concepts C while the teacher's map contains a concept c_2 that is mostly connected to most of the concepts in C, then the student may be confusing c_2 with c_1. In this case, c_1 is said to be a confused concept. If a student incorrectly links a concept c_1 to a set of concepts C, while in the teacher's map, a concept c_2 is connected to the concepts in C, then it can be suggested to the student that c_1 may have to be substituted to c_2. Linkage analysis can also identify less obvious misconceptions. For example, when a concept c is correctly linked to other concepts in a set C, but the concepts in C are incorrectly linked, then the student may have misunderstood c in the first place.

A set of algorithms has been developed to perform the above form of linkage analysis automatically for given student and teacher maps. Its purpose is to provide automated support for assessment in concept map based e-learning.

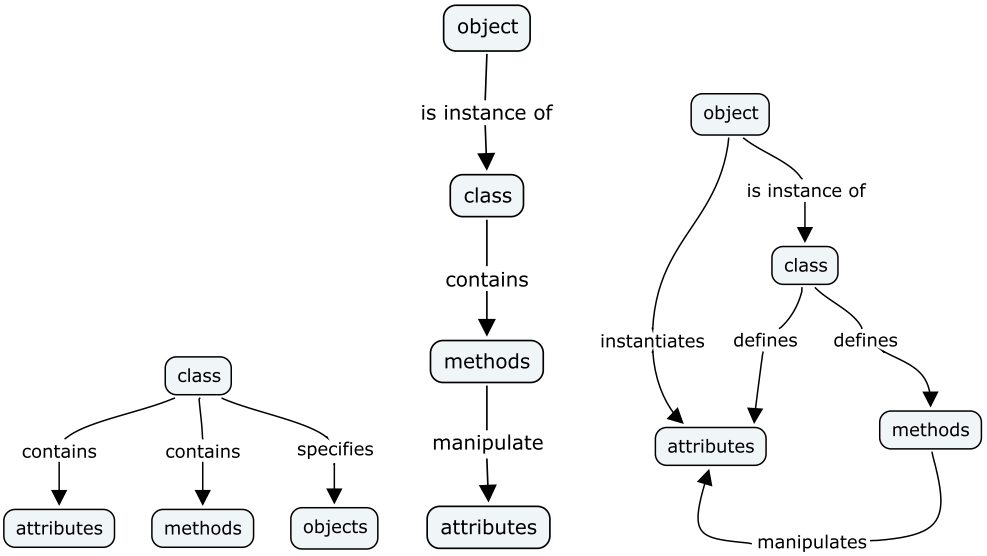
2.2.2 Spoke, chain, net differentiation

Kinchin and Hay [Kinchin and Hay2000] propose to extract from concept maps, three types of substructure: spokes, chains and nets. In essence, a *spoke* is equivalent to a single level hierarchy, a *chain* corresponds to a sequence of concepts and a *net* denotes a substructure where a pair of concepts can be related to one another by means of different sets of concept-links. Figure 2 illustrates these substructures by means of archetypical concept maps containing the same four computer programming concepts.

Spoke, chain and net substructures classify how well certain concepts are integrated in a learner's mental models of the subject of study. Indeed, the substructures prescribe if and how a learner's concept map collapses under the influence of new information that contradicts it.

In a spoke substructure, the learner has identified certain concepts that are related to a given core (or key organising) concept, but fails to identify how the former concepts are related to one another. As such, they may be unable to relate concepts to one another in situations that do not include the core concept. For example, the learner of the concept map of Figure 2(a) would be unable to specify what the attributes of a given object are, without referring to the object's class.

A chain substructure is usually an indication of rote learning, as the sequence depicted by chain often corresponds to the order in which concepts were introduced in the lecture. Certain links in such substructures are inevitably tenuous, and may break down when confronted with new information. For example, when confronted with certain methods that do not manipulate attributes, the learner of the concept map shown in Figure 2(b) may be left wondering how attributes are



(a) Spoke (b) Chain (c) Net

Figure 2: Spoke, Chain and Net structures

related to objects.

In a net substructure, concepts are integrated with one another more strongly. Therefore, such substructures are more robust to contradictory information than spoke and chain structures. For example, the learner of the concept map of Figure 2(c) would not be expected to experience difficulty with the aforementioned sample problems. Evidence suggests that net structures indicate meaningful learning [Kinchin et al.2005].

2.2.3 Qualitative simulation

Qualitative simulation refers to a set of techniques devised to extrapolate the behaviour of physical systems in terms of qualitative descriptions. Like numerical simulation, it formalises system behaviour by means of mathematical models. But, the quantities that model's variables take over time are denoted using crude qualitative distinctions, such as "above zero", "up to a local maximum" and "decreasing".

Biswas et. al. [Biswas et al.2005] have devised a method to use qualitative simulation for the assessment of causal concept maps, i.e. concept maps in which all links describe causal relations with a specific pre-defined semantics. The approach requires a somewhat narrowly defined concept mapping task. It involves the students in teaching an autonomous agent (i.e. an independent problem solving computer program), known as "Betty's Brain", about a particular type of system (such as a river's eco-system) by defining the agent's mental model by means of a causal concept map. The agent is then quizzed by a series of questions that require it to predict certain effects of changes in the system under investigation. If its predictions are invalid, another agent interacts

with the student to explain the predications of "Betty's Brain" and to help diagnose the error.

3 Concept map assessment in teaching computer programming

3.1 Evaluation criteria

Assessment methods are normally evaluated using a range of criteria. Some of these criteria, such as fairness and transparency, refer to the environment in which the assessment occurs. This paper will not consider such issues, focussing instead on the criteria that are primarily affected by the choice of assessment method and the subject of assessment. These are validity, reliability and efficiency. Table 3 summarises who this discussion can be applied to the assessment methods discussed herein.

3.1.1 Validity

The validity of an assessment method is the extent to which the measurements it produces are accurate reflections of what the method intends to determine. Intuitively, a valid assessment method is said to measure the right thing.

The concept map assessment methods surveyed herein appear to measure very different aspects of concept maps, ranging from the equivalences with a single concept map to broad structural features of concept maps. In empirical studies of quantitative methods, a method's validity is normally defined based as its correlation with another assessment method [McClure et al.1999, West et al.2002]. While this approach allows for the significance of the results to be validated, it can be flawed, especially if the validity of the method that others are compared against is not demonstrated. In this paper, a more qualitative approach will be taken by identifying whether the concept map features that the assessment methods examine are important to achieve the learning outcomes.

One feature that is particularly relevant in this respect is the amount of variability of correct concept maps that the assessment method tolerates. This varies between disciplines and sometimes, within disciplines. Biglan defines a discipline's hardness as the degree to which it contains a central body of theory that is universally accepted within its membership [Biglan1973]. As such, methods that define valid concept maps more narrowly are more suitable for hard disciplines while methods that employ looser definitions are more suitable for soft disciplines.

The selection of application domains for the assessment methods surveyed herein confirms this hypothesis. Those that are primarily applied to disciplines classified to be hard in Biglan's framework tend to classify relations between concepts into correct and incorrect ones. Indeed, all

of the quantitative assessment methods discussed herein have been applied primarily to (hard) science education. Those with substantial applications in soft disciplines do not impose such precise criteria. For example, the qualitative simulation approach of Biswas et. al. has been primarily applied to ecological modelling [Biswas et al.2005], which is a discipline that has reached little consensus regarding the theories it has developed. Clearly, assessment methods that rely on comparing a student concept map with a model or teacher concept map (e.g. the closeness index and linkage analysis) are entirely unsuitable for such disciplines. Also, techniques that rely on scoring the relevance of links between concepts may be difficult to apply validly to maps of soft concepts given that soft domain allows for more valid permutations of map structures.

Surveys suggests that most computer scientists consider their discipline to be a hard one [Clark2003]. However, crucial sub-disciplines of software engineering, such as human-computer interaction and systems requirements analysis, are soft disciplines [Dix et al.2004]. For example, usability principles of user interfaces, such as predictability and consistency are difficult to define formally. They are recognised intuitively when confronted with an example of a user interface that exhibits these features. However, they may mean different things to different people. As such, concept maps of these aspects of computer programming correspond to mostly personal interpretations and, therefore, they may be assessed most effectively by methods that measure the maps' sophistication rather than their similarity to a given model.

3.1.2 Reliability

The reliability of assessment methods refers to the consistency of the method between different raters. In empirical studies, it is normally defined as interrater correlation. Some of the assessment methods discussed herein have already been automated and most can at least be formalised by means of an algorithm and are, therefore, automatable. Obviously, when an assessment method is applied by a machine, reliability is perfect. Three methods, the holistic, structural and relational scoring methods can not be automated. For these methods, research suggests that the reliability of the holistic method is rather poor while that of the structural and relational scoring method varies between studies and application domains [McClure et al.1999, West et al.2002].

3.1.3 Efficiency

The efficiency of an assessment method is its resource-economy. The main resources required for concept map assessment are staff time for preparation of the concept mapping task and assessment and the actual assessment. Firstly, the preparation time can be substantial when a model concept map needs to be designed. Some methods, such as the closeness index and linkage analysis, require this. Other methods, such as the holistic, structural and relational scoring methods, benefit from a model concept map [McClure et al.1999]. Secondly, the assessment time can be negligible those

assessment methods that are automatable. For the assessment methods that are not automatable, there is only little research into the relative efficiency of these approaches. Work by McClure et. al. suggests that the holistic and structural methods are somewhat more efficient than the relational one, though these results are specific only to one concept mapping task [McClure et al.1999].

3.2 Applications

The potential applications of concept mapping in teaching computer programming are threefold. They can be employed to assess:

- *Knowledge of theoretical computer programming concepts.* Programming languages employ a small number of mathematical concepts that students find difficult to understand [Hu2006]. These concepts have very precise definitions, allowing for little room for interpretation. For example, the basic programming concepts of "class", "object", "attribute" and "method" have precise meanings and the number of ways in which they can be related to one another is limited, as illustrated in Figure 2. Given the small size of this domain, most students will be aware of most of the relevant concepts and consequently, research suggests that the symptoms of student misconception of this knowledge takes to form of subtle flaws in the concept maps, such as minor structural deviations from the correct definition [Liu et al.2005].

  We hypothesise that approaches that can compare teacher and student maps are more likely to produce useful information to diagnose student misconceptions. The closeness index and linkage analysis provide the most detailed assessment methods to perform such comparisons. Both approaches have been automated, but they require the construction of a model. However, the effort required to produce such maps is expected to be limited as most programming courses are restricted to a single language and number of distinct theoretical concepts in each programming language is relatively limited.

- *Ability to use and form a synthesis of software libraries.* While most modern software is large and complex, many software products use similar components. For example, most user interfaces of software use windows, icons, menus and pointers that are similar in appearance and behaviour. Such components are available through software libraries. Therefore, computer programmers need to be able to employ effectively large software libraries that implement these components. Although the relevant entities and their interactions are defined precisely for each software library in a so-called Application Programming Interface (API), effective programmers do not memorise them. Instead, they develop their own conceptual models of APIs, which enable them to quickly look up in documentation, the programming instruction for the functionality they require in their programs.

Assessment method	Validity		Reliability	Efficiency	
	Permitted variability	Objective of measurement		Preparation time	Assessment time
Holistic scoring	unspecified	unspecified	low	teacher map is beneficial	moderate (?)
Structural scoring	moderate	map structure and labels	good	teacher map is beneficial	moderate (?)
Relational scoring	moderate	individual relations and labels	good	teacher map is beneficial	moderately high (?)
Closeness index	very low	similarity to model map	automated	teacher map is required	automated
Linkage analysis	very low	similarity to model map	automated	teacher map is required	automated
Spoke, chain, net differentiation	high	map structure	automatable	task assignment only	automatable
Qualitative simulation	moderately	model predictions	automatable	prediction assignments and solutions	automatable

Table 3: Overview of features affecting the validity, reliability and efficiency of concept map assessment method. (?) indicates inconsistent results in the literature.

Student concept maps can reveal how effectively students will be able to navigate API documentation. Different types programming problem require that components be combined in different ways. Hence, in order to be able to solve a wide range of problems, a programmer should be aware of alternative ways of relating API concepts to one another. It is expected that such knowledge manifests itself in the form of more sophisticated structures of API concept maps.

Therefore, we hypothesise that concept map assessment methods that analyse the structural complexity of concept maps would be the most appropriate means of assessing this ability. Chain-spoke-net differentiation is designed to assess precisely these types of structural sophistication and, therefore, is expected to be a valid approach to assessing concept maps on this topic. This approach is also automatable and independent of a model concept map, which suggests that it would also be reliable and efficient in this setting.

- *Ability to construct concept maps.* Programmers are not only required to assimilate technological and mathematical knowledge. They also need to be able to learn the relevant knowledge about the domain which the software they develop is to be integrated. The crucial part of this learning process is modelling the domain knowledge. To that end, programmers routinely communicate with diagrammatic formalisms such as entity relationship diagrams, state transition diagrams and interaction diagrams. These different types of diagram can be conceived to be highly formalised versions of concept maps. Therefore, the ability to draw concept maps is a particularly useful skill for computer programmers.

  Assessing software application specific concept maps is difficult. The purpose of these diagrams is not necessarily completeness, accuracy or structural sophistication. Their quality depends on how well they reflect an application's requirements. An effective way of testing this criterion is to explore the logical implications of the conceptual model for specific test cases. This approach has been proposed in the qualitative simulation based method, though for the restricted setting of causal models. Such an assessment method would match the learning outcomes of the modelling exercise and as substantial parts of the approach can be automated, with the important exception of test case generation, it is expected to be reliable and efficient as well. The development of such an assessment approach would therefore present a potentially useful piece of future research.

4 Conclusion

This paper has presented a survey of concept map assessment methods and examined their suitability in the context of teaching computer programming, with a particular focus on validity, reli-

ability and efficiency of these methods. The survey has identified seven concept map assessment methods that differ considerably from one another: holistic scoring [McClure et al.1999], structural scoring [Novak and Gowin1984], relational scoring [McClure et al.1999], the closeness index [Goldsmith et al.1991], linkage analysis [Liu et al.2005], chain-spoke-net differentiation [Kinchin and Hay2000] and qualitative simulation [Biswas et al.2005]. While application domains exist for each method that are particularly well suited to it, three applications of concept mapping in computer programming teaching have been identified and corresponding assessment method have been proposed. Firstly, the closeness index and linkage analysis were suggested as suitable assessment methods for determining a student's understanding of a programming language's basic concepts. Secondly, chain-spoke-net differentiation was put forward as an effective method to evaluate a student's awareness of software libraries. Thirdly and finally, a qualitative simulation based approach was proposed to assess student's model building ability.

It should be recognised that the work presented herein was not underpinned by empirical data, other than the work developed by other authors. This paper is intended to be a critical literature review, aimed at inspiring future work. The authors plan to employ concept mapping in the context of an intermediate computer programming course, which aims to teach students to implement graphical user interfaces in Java and to employ Human-Computer Interaction principles in the design of graphical user interfaces.

Acknowledgements

The first author was partly support by the Nuffield foundation grant NAL/32730. Both authors are grateful for the important suggestions received from the anonymous referees, whilst taking full responsibility for the views expressed in this paper.

References

[Ausubel1968] Ausubel, D. (1968). *Educational psychology: A cognitive view.* Holt, Rinehart & Winston.

[Biglan1973] Biglan, A. (1973). The characteristics of subject matter in different acadamic areas. *Journal of Applied Psychology*, 57:204–213.

[Biswas et al.2005] Biswas, G., Leelawong, K., Schwartz, D., Vye, N., and at Vanderbilt, T. T. A. G. (2005). Learning by teaching: a new agent paradigm for educational software. *Applied Artificial Intelligence*, 19:363–392.

[Bransford et al.1999] Bransford, J., Brown, A., and Cocking, R., editors (1999). *How people learn: Brain, mind, experience, and school*. National Academy Press.

[Clark2003] Clark, M. (2003). Computer science: a hard-applied discipline? *Teaching in Higher Education*, 8(1):71–87.

[Dix et al.2004] Dix, A., Finlay, J., Abowd, G., and Beale, R. (2004). *Human-Computer Interaction*. Prentice-Hall.

[Goldsmith et al.1991] Goldsmith, T., Johnson, P., and Action, W. (1991). Assessing structural knowledge. *Journal of Educational Psychology*, 83:88–96.

[Hay et al.2007] Hay, D., Wells, H., and Kinchin, I. (2007). Quantitative and qualitative measures of student learning at university level. *Submitted for Journal Publication*.

[Hu2006] Hu, C. (2006). It's mathematical, after all - the nature of learning computer programming. *Educational Information Technology*, 11:83–92.

[Kinchin et al.2005] Kinchin, I., DeLeij, F., and Hay, D. (2005). The evolution of a collaborative concept mapping activity for undergraduate microbiology students. *Journal of Further and Higher Education*, 29(1):1–14.

[Kinchin and Hay2000] Kinchin, I. and Hay, D. (2000). How a qualitative approach to concept map analysis can be used to aid learning by illustrating patterns of conceptual development. *Educational Research*, 42(1):43–57.

[Liu et al.2005] Liu, C.-C., Don, P.-H., and Tsai, C.-M. (2005). Assessment based on linkage patterns in concept maps. *Journal of Information Science and Engineering*, 21:873–890.

[Mayer1981] Mayer, R. (1981). The psychology of how novices learn computer programming. *ACM Computing Surveys*, 13(1):121–141.

[McClure et al.1999] McClure, J., Sonak, B., and Suen, H. (1999). Concept map assessment of classroom learning: reliability, validity, and logistical practicality. *Journal of Research in Science Teaching*, 36(4):475–492.

[Novak2005] Novak, J. (2005). Results and implications of a 12-year longitudinal study of science concept learning. *Research in Science Education*, 35:23–40.

[Novak and Gowin1984] Novak, J. and Gowin, D. (1984). *Learning how to learn*. Cambridge University Press, New York.

[Ruiz-Primo and Shavelson1996] Ruiz-Primo, M. and Shavelson, R. (1996). Problems and issues in the use of concept maps in science assessment. *Journal of Research in Science Teaching*, 33:569–600.

[West et al.2002] West, D., Park, J., Pomeroy, J., and Sandoval, J. (2002). Concept mapping assessment in medical education: a comparison of two scoring systems. *Medical Education*, 36:820–826.