Concept Maps: Theory, Methodology, Technology
Proc. of the First Int. Conference on Concept Mapping
Pamplona, Spain 2004

A NEW APPROACH TO MEANINGFUL LEARNING ASSESSMENT USING CONCEPT MAPS: ONTOLOGIES AND GENETIC ALGORITHMS

Francisco Edson Lopes da Rocha, Júlio Valente da Costa Jr & Eloi Luiz Favero,
Universidade Federal do Pará, Brasil
Email: felr@ufpa.br, www.inf.ufpa.br/~edson

Abstract. In this article we discuss the use of concept maps (CMs) in learning assessment. As an alternative to the comparison of the learner's CM with the teacher's CM, in order to certify what is right or wrong or to attribute a grade, we present a new approach to assess CMs: we consider learning assessment as an adaptive and evolutionary problem and we show how to use ontologies and machine learning, through genetic algorithms (GAs), to assess CMs. The ontologies we use store knowledge, in the form of concepts and propositions, and functions to measure the semantic distance between CMs. The GA, using the ontology, generates the search space (collections of CMs) used to show learners the alternatives to their possible faults when learning concepts and propositions. The complete assessment of a CM includes the analysis of its hierarchical structure, the recognition of learning types, and the analysis of semantic similarity with the CMs of the search space. We also show the actions executed by the GA to construct the CMs of the search space and the concepts present in the ontology and ignored by the learner.

1 Introduction

Concept mapping is a process of knowledge construction. The concept maps (CMs) that result from this process represent significant combinations between concepts in the form of propositions – a concept is a regularity perceived in objects, events, situations or properties (Novak, 1998, p.22), or concepts are the abstracted criterial attributes that are common to a given category of objects, facts, or phenomena (Ausubel, 2000, p.2). A proposition is a combination of two or more concepts mediated by linking words (Novak, 1998, p.32).

The linking words that appear in each proposition denote *values* assigned to binary relations that exist between concepts combined in a proposition. They can be organized in hierarchies of types (super-types) considered as metadata (Fischer, 2001). For example, in the propositions <TREE has ROOT> and <TREE feeds through ROOT> the linking words <has> and <feeds through> can be seen as values of the super-types **partition** and **process.**

CMs are established on three fundamental theoretical principles: hierarchical structure, progressive differentiation, and integrative reconciliation (Novak & Gowin, 1999). The hierarchical structure of a CM is based on the concept of *inclusion*, that is, each concept put in a CM has a level of inclusion relative to the other concepts already in the map. Therefore, inclusion is used to classify the concepts of a CM and is fundamental for the construction of meanings (Costa Jr et al., 2004). Figure 1 shows a CM as proposed by Novak. <HUMAN LEARNING> is the concept of higher degree of inclusion and classifies the concepts <COGNITIVE LEARNING>, <AFFECTIVE LEARNING>, and <PSYCHOMOTOR LEARNING>, according to a **characteristic** (attributes) dimension.

Progressive differentiation is the process of meaningful learning in which learners increase the degree of elaboration of a concept as they increase their understanding about it (Ausubel, 2000). In order to detect progressive differentiation in a CM, it is necessary to observe if the concept object of the learning classifies other concepts in one or more dimensions. In the example illustrated in Figure 1, the concept <COGNITIVE LEARNING> is progressively differentiated.

Integrative reconciliation is the meaningful learning in which the learner discerns relations between concepts not initially categorized. There are two types of integrative reconciliation learning: *superordinate integrative reconciliation* and *combinatorial learning.* Superordinate integrative reconciliation occurs when the learner identifies a more inclusive concept, not initially present in the CM, which includes concepts already learned and represented in the map. In order to observe superordinate integrative reconciliation, it may be necessary to observe two distinct moments of the learning represented in the CMs and to have specific graphical notation for it (see Figure 2).

Combinatorial learning occurs when the learner, without identifying a more inclusive concept, can discern the need of relating concepts placed in different branches of the same CM. Figure 2 presents two situations of integrative reconciliation: in Figure 2(a), the concept <KNOWLEDGE> superordinates <CONCEPTS> and <PROPOSITIONS>, and these two combine to express how prepositions are formed.

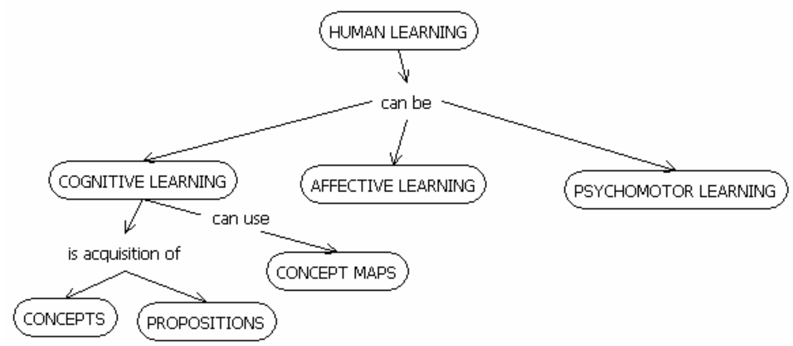

Figure 1. Concept Map about HUMAN LEARNING

Figure 2(b) differs from Figure 2(a) in an important point: the proposition notation with solid lines informs that the type of learning in Figure 2(a) is *progressive differentiation* while the notation with dashed lines in Figure 2(b) informs that the learning type is integrative reconciliation.

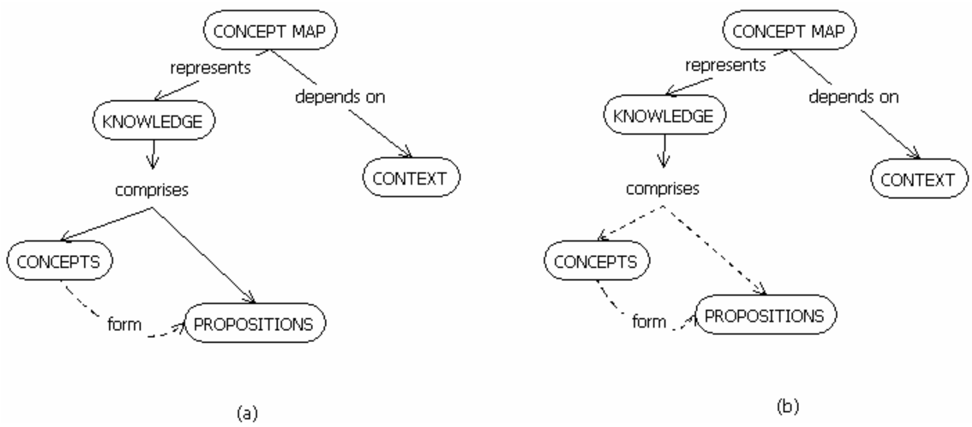

Figure 2. Combinatorial integrative reconciliation and superordinate integrative reconciliation

Assessing learning in an educational context can be seen as the process of characterizing what a student knows (Turns et. al., 2000). Using CMs to assess learning is important because their structure contains the cognitive elements considered evidence of learning in Ausubel and Novak's theory: concepts, hierarchy of concepts, propositions, progressive differentiation, combinatorial and superordinate integrative reconciliation. However, constructivist learning can lead to many different ways of constructing the same knowledge. For example, innumerous types of relations can exist between <PLANT>, <ROOT>, <STEM>, <LEAF>, <FRUIT>, <SEED>, and <FLOWER>. Learning assessment of these concepts based on a single reference MC would be very inaccurate, because a single CM is not capable of encompassing all the potential learning situations involved. On the other hand, the task of constructing reference CMs capable of coping with this process of individual assessment would be overwhelming. An alternative to face this problem is to provide the teacher with the possibility of creating domain ontologies with the concepts and their relations in the context of a learning task. A complementary mechanism could then simulate all the possible learning processes and represent them as collections of CMs. This is the fundamental idea why we chose to use genetic algorithms as the mechanism capable of producing these collections.

In this article we present the CMTool learning assessment system (Rocha & Favero, 2004). CMTool's main objective is to provide support to the practical application of Ausubel and Novak's theory in classroom, focusing on the assessment of learning. The assessment process in CMTool is based on: (i) a domain ontology that stores concepts, binary relations, linking words, and the function to measure semantic distances between concepts, propositions and CMs; (ii) a genetic algorithm (GA) that, based on the ontology data, generates collections of CMs (search space) for the accomplishment of the assessment; (iii) an assessor that uses the search space generated by the GA and the ontology to detect evidences of learning in the learner's CM.

The article contains five sections, including this introduction. In section 2, a general description of CMTool is presented, including the assessment system. In section 3, we present an example of assessment of a simple CM. Section 4 presents research related to our work. We conclude the article in section 5.

2 The CMTool Environment

Figure 3 presents CMTool’s block diagram, which encompasses five modules: administrator; concept map editor; ontology editor; learning assessor; genetic algorithm (GA); and a repository (user information, instances of search spaces generated by the GA, ontologies, and instances of CMs of users registered in the environment).

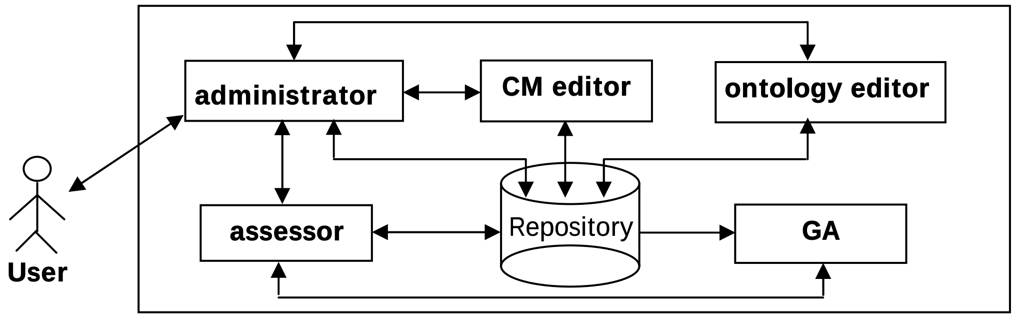

Figure 3. Architecture of the CMTool environment

The administrator is responsible for controlling environment access (registration and identification of users, for example), and for allowing access to the editors and to the assessor. The registration system of the administrator assigns to the users the right of accessing specific functionalities of the environment. For example, teachers registered in the environment can access the CM editor, the ontology editor, and the assessor. Students, on the other hand, are able to access only the CM editor. The CM editor implements a visual language for constructing CMs in compliance with the principles of Ausubel and Novak’s meaningful learning theory. The module provides a graphical interface composed of a drawing panel and a toolbar with the metasigns and the functionalities necessary to draw, edit, save, and recover concept maps. Table 1 presents some of the signs of the CM editor.

Sign	Semantics
PLATYPUS	Used to represent generic concepts (e.g., PLATYPUS)
PHILLY	Used to represent examples of concepts (e.g., PHILLY)
— link-word →	Used to represent hierarchies and progressive differentiation
- - - - link-word - - >	Used to represent hierarchies and integrative reconciliation
<<>>	Used to write grouped propositions (conjunction)

Table 1. CMTool environment signs

In order to write concepts, the visual language provides two symbols: a rectangle with rounded corners and a rectangle with straight corners. The first one is used to represent generic concepts, while the second one is used to represent examples. The lines of the arrows that link concepts can be dashed or solid, straight or curved. The style of the line (dashed or solid) is used to clarify the type of learning (progressive differentiation or integrative reconciliation), while the form of the line (straight or curved) is only a convenience for the drawing process. The *and* conjunction is a symbol without graphical representation. It is used to compact the writing of propositions starting in the same concept, when they use the same linking word. In Figure 2(a) the linking word *comprises* is written on an *and* conjunction. In this case, the reading of the propositions starting in <KNOWLEDGE> could be: <<KNOWLEDGE comprises CONCEPTS> and <KNOWLEDGE comprises PROPOSITIONS>> or <<KNOWLEDGE comprises CONCEPTS and PROPOSITIONS>>.

Examples are elements that must be distinguishable in a CM, either to help grading it, or to detect more precise understanding during the learning process (Novak & Gowin, 1999, p.53). That is the reason why some CM editors use more than one symbol to represent concepts (for example, see Cunha & Fernandes, 2002). In CMTool, the graphic symbols of the environment are part of a visual language (not detailed in this paper), with syntax and semantics. As in all languages, some users will be less experienced than others. Consequently, we do not expect young learners to use it in all its potential. Nevertheless, our experience with graduate and undergraduate students showed us that progressive differentiation and integrative reconciliation, for example, can occur consciously during learning. As a result, it is necessary to distinguish them graphically, from an external point of view, and computationally, from an internal point of view.

The ontology editor is used to construct and store in the repository the domain ontologies that correspond to the learning tasks used to organize the teaching of the topics of a discipline. An ontology is a catalogue of *types* *of things* for a determined domain of interest D from the perspective of a language L constructed with the purpose of describing D (Sowa, 2000). In CMTool, the types in the ontology represent concepts and linking words that can be described by the visual language of the environment.

The functioning of the GA is centered in the ontology. Genetic algorithms are blind search algorithms whose purpose is to find a collection of possible results for a given problem. Each result in the collection is an individual and the collection is a population (Zbigniew & Fogel, 2002). In CMTool, a population is a set of CMs with characteristics similar to those of a CM presented as input to the algorithm. Each CM in the generated population has its history of construction, which registers the meaningful learning simulation process used by the GA during the construction of the associated CM.

The assessor uses the results produced by the GA and the ontology to produce a complete assessment of the learning of a student or group of students. The assessor applied to a single CM can inform: (i) valid propositions; (ii) valid concept hierarchies; (iii) valid integrative reconciliations; (iv) valid progressive differentiations; (v) examples; (vi) the relation of semantic proximity between the CM constructed by the learner and those generated by the GA; and (vii) the collection of actions that were used by the GA to generate the population. The assessor applied to two or more CMs can detect if there was evolution in the learning.

The repository stores CMs, ontologies, assessments and populations of CMs generated by the GA. The ontologies are generated by the ontology editor. These ontologies will be read by the GA in order to generate the populations of CMs, and also by the assessor, that obtains the result of the functions that calculate semantic distances. A population of CMs is stored in the repository by the assessor after the accomplishment of an assessment. Each population stored in the repository is identified by the ontology and the concepts used to build it. When receiving a CM to analyze, the assessor examines the repository in search of a population that can be used in the current analysis. If the assessor finds a population, it recovers and uses it. If not, it activates the GA to generate the desired population. The assessor also records in the repository the result of each assessment accomplished.

3 An assessment example

In this section we present an example that shows the functioning of the GA and of the assessor of the environment. The example is related to the learning task *What it is a plant?*

3.1 Data stored in the ontology

For this learning task, the ontology has the following data:

1. concepts: {PLANT, ROOT, STEM, LEAF};
2. relations: partition type: {has part, contains, is example, is instance};
3. binary relations: {<PLANT, r_1, ROOT>, <PLANT, r_2, STEM>, <PLANT, r_3, LEAF>};
4. values of r_1, r_2, r_3: {has part, contains}.

With this information it is possible to determine all the valid propositions for this context, as well as the semantic distances between them, as follows. Valid propositions: $\{p_1, p_2, p_3, p_4, p_5, p_6\}$ = {<PLANT, has part, ROOT>, <PLANT, contains, ROOT>, <PLANT, has part, STEM>, <PLANT, contains, STEM>, <PLANT, has part, LEAF>, <PLANT, contains, LEAF>}; semantic distances between the propositions: $d_p(p_1,p_2) = d_p(p_3,p_4) = d_p(p_5,p_6) = 0$; $d_p(p_1,p_i) = \infty, 3 \le i \le 6$; $d_p(p_2,p_i) = \infty, 3 \le i \le 6$.

The semantic distance between propositions is symmetric ($d_p(p_i,p_j) = d_p(p_j,p_i)$). The distance $d_p(p_i,p_j) = 0$ means that the propositions p_i e p_j are semantically equivalent, while $d_p(p_i,p_j) = \infty$ means that the propositions p_i e p_j are semantically incomparable. The function implemented also admits $d_p(p_i,p_j) = 1$ for semantically similar propositions (same type of binary relation, but the value assigned to the relation is not meaningful in the context of the learning task, according to the ontology). The semantic distance between CMs is defined as a function of the semantic distance of the propositions in the maps that can be compared (encompassing the same concepts). For example, d_p(<PLANT, has part, ROOT>,<PLANT, contains, ROOT>) = 0, because <has part> and <contains> are equally included in the **partition** type describing the relation <PLANT, r_1, ROOT>. On the other hand, d_p(<PLANT, has part, ROOT>,<PLANT, is instance, ROOT>) = 1 and d_p(<PLANT, has part, ROOT>,<PLANT, has characteristic, ROOT>) = ∞. Given CM_1 = {<PLANT, has part, ROOT>,<PLANT, has part, STEM>} and CM_2 = {<PLANT, has part, ROOT>,<PLANT, contains, STEM>}, $d_m(CM_1, CM_2) = 0$, because max((d_p(<PLANT, has part, ROOT>,<PLANT, has part, ROOT>), d_p(<PLANT, has part, STEM>,<PLANT, contains, STEM>)) = 0, where *max* is the biggest value in the argument list of the function.

3.2 Actions of the GA

Figure 4 represents actions of the GA. Based on the CM of the learner submitted to the assessor (Figure 4(a)) and on the ontology for the learning task underway, the GA determines the initial population of CMs (Figure 4(b)). Afterwards, the GA selects in the current population the individuals better fitted to be the parents of the next population (in the example, all the individuals are selected because they all have the same fitness value). It then submits these individuals to the genetic operators of crossing and mutation (Figures 4(c) and 4(d)). The new population is formed by the ancestors and their descendants. The algorithm repeats this procedure until n populations (n is configurable) are generated or all the individuals of a population reach a satisfactory fitness value. Figure 4(e) represents the final population generated by the GA for the learning session in progress.

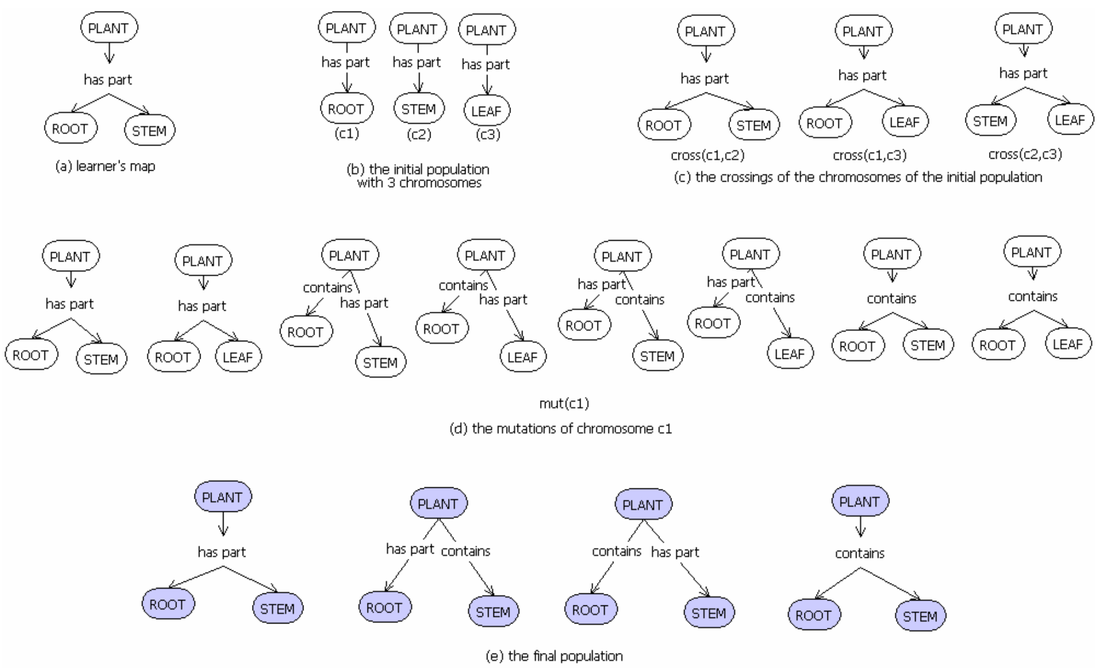

Figure 4. Actions of the GA

The GA is based on axioms that describe how to form its basic elements and genetic operators. The basic elements are *base*, *gene*, *chromosome*, and *population*, and the genetic operators are *crossing* and *mutation*. The chromosomes (CMs) of a new population are those with fitness equal or greater than the average fitness of the previous population. The average fitness of a population is the arithmetic mean of the fitness of its chromosomes. The fitness of a chromosome is the weighted sum of the fitness of each of its genes, and the fitness of a gene is calculated based on the binary relation it expresses. The taxonomy that allows assigning values to binary relations is described in Costa Jr et al. (2004). For example, considering the genes g_1 = <PLANT, has part, LEAF> and g_2 = <PLANT, has part, ROOT> (see Figure 4(b)), their fitness equals 2, because both belong to the ontology. The weight of g_1 is 1, because only the concept <PLANT> exists in CM_{ap} (see Figure 4(a)), while the weight of g_2 is 2, because the concepts <PLANT> and <ROOT> exist in CM_{ap}. Therefore, the chromosome <<PLANT, has part, LEAF>, <PLANT, has part, ROOT>> has a fitness degree of 6 (2*1 + 2*2). The weights are the mechanism used to privilege the concepts present in CM_{ap}. The complete specification of the GA used in CMTool is in Rocha et al. (2004).

3.3 Assessment Results

Figure 5 presents the results of an assessment accomplished by CMTool. The results are organized in four parts: (a) Hierarchical structure and learning types demonstrated; (b) Semantic similarity between the assessed CM and the CMs generated by the GA; (c) Actions necessary for the reconstruction of the generated CMs; (d) Omissions in the assessed CM.

Assessment Results

a) Hierarchical structure and learning types

01. CM Assessed: CM_{ap} = {<PLANT,has part,ROOT>,<PLANT,has part,STEM>}
    Concepts: {PLANT,ROOT,STEM}
    Propositions: {<PLANT,has part, ROOT>(p_1),<PLANT,has part,STEM>(p_2)}
02. Hierarchical level:
    Level 0: PLANT
    Level 1: ROOT,STEM
03. Valid hierarchies(1): {<PLANT,partition,ROOT>,<PLANT,partition,STEM>}
04. Invalid hierarchies(0): {}
05. Valid propositions(2): {<PLANT,has part,ROOT>,<PLANT,has part,STEM>}
06. Invalid propositions(0): {}
07. Valid integrative reconciliations(0): {}
08. Invalid integrative reconciliations(0): {}
09. Valid progressive differentiations(1): {<<PLANT>,partition,<ROOT,STEM>>}
10. Invalid progressive differentiations(0): {}
11. Examples(0): {}

b) Semantic similarity

01. Population generated by the GA(4):
    Concepts {PLANT,ROOT,STEM}
    Ontology: EN250-01;
    CM_1 = <<PLANT,has part,ROOT>(p_3),<PLANT,has part,STEM>(p_4)>
    CM_2 = <<PLANT,has part,ROOT>(p_5),<PLANT,contains,STEM>(p_6)>
    CM_3 = <<PLANT,contains,ROOT>(p_7),<PLANT,has part,STEM>(p_8)>
    CM_4 = <<PLANT,contains,ROOT>(p_9),<PLANT,contains,STEM>(p_{10})>
02. Calculation of semantic distances:
    CM_{ap} versus CM_1:
        $d_m(CM_{ap},CM_1) = 0$; (CM_{ap} and CM_1 have the same meaning)
    CM_{ap} versus CM_2:
        $d_m(CM_{ap},CM_2) = 0$; (CM_{ap} and CM_2 have the same meaning)
    CM_{ap} versus CM_3:
        $d_m(CM_{ap},CM_3) = 0$; (CM_{ap} and CM_3 have the same meaning)
    CM_{ap} versus CM_4:
        $d_m(CM_{ap},CM_4) = 0$; (CM_{ap} and CM_4 have the same meaning)

c) Actions for the reconstruction of the generated CMs

CM_1:
    form the propositions: {p_3, p_4};
    combine: p_3, p_4 (differentiate PLANT progressively)
CM_2:
    form the propositions: { p_5, p_6};
    combine: p_5, p_6 (differentiate PLANT progressively)
CM_3:
    form the propositions: { p_7, p_8};
    combine: p_7, p_8 (differentiate PLANT progressively)
CM_4:
    form the propositions: { p_9, p_{10}};
    combine: p_9, p_{10} (differentiate PLANT progressively)

d) Ontology concepts absent in CM_{ap}:

{LEAF}

Figura 5. Results of a CM assessment accomplished by CMTool

Part (a) reports if the concepts used in the CM submitted for assessment (CM_{ap}) are related to the learning task underway, as registered in the ontology used, and if the learner's propositions are valid in this context. The inclusion level of each concept is verified with the help of a semantic reading by inclusion level of CM_{ap} (Rocha & Favero, 2004). The result of this reading of CM_{ap} is compared to the result of an identical reading performed on the CMs generated by the GA. If the results match, the assessor certifies the correction of the hierarchical levels present in CM_{ap}. The assessor also verifies if the inclusion of the concepts is made through correct classification types. For example, in <PLANT, partition, ROOT> <PLANT> classifies <ROOT> based on the partition type (whole-part), which is confirmed by the ontology. Other results of part (a) can be seen in Figure 5.

Part (b) presents the semantic comparison of CM_{ap} to the final population of CMs generated by the GA. The objective is to present to the learner other valid forms of mapping the knowledge represented in the ontology of the learning task underway. The assessor calculates the semantic distance between CM_{ap} and each one of the CMs in the final population generated by the GA. If any of the calculated values is different from zero, detailed information containing the possible alternatives to the identified misconception are presented to the learner.

Part (c) details the actions that were taken by the GA to construct the population of CMs presented in part (b). The objective is to show to the learner how to construct forms of knowledge representation alternative to his/her own (presented in part (a)). Finally, part (d) presents the list of concepts that, although present in the ontology used, were not mapped by the learner. The list may indicate the need for reinforcement of specific topics of the discipline.

4 Related research

The computerized treatment of CMs as support to **meaningful learning** has been the focus of many research projects. Concerning map editors, we have found excellent tools like CmapTools (Cañas et al., 1999) and LifeMap (http://www.robertabrams.net/conceptmap/lifemaphome.html). They provide several possibilities of construction and sharing of knowledge via CMs.

Concerning learning assessment using CMs, some ongoing research projects need to be mentioned. Araújo et al. (2003) analyze the work overload on the teacher resulting from the use of CMs in the process of learning and propose an automated way to minimize this problem. Their assessment proposal is based on the comparison between the learner's CM and the teacher's CM. Cunha & Fernandes (2002) propose a cooperative environment to intermediate the synchronous interaction among learners and the learning facilitator. In this environment, designed to run on the Web, the assessment is based on the comparison between a reference CM and the learner's CM. Cabral & Giraffa (2002) also propose learning assessment based on the comparison between the learner's CM and the teacher's CM. Chung et al. (2002) developed a prototype in which learning assessment is based on the comparison with CMs constructed by specialists. McGriff (2001) describes how to use CMs to measure a learner's cognitive structure with the help of computational procedures. Turns et al. (2000) proposes the use of CMs to assess learning based on the grading of different characteristics of the map, such as width, depth, and connectivity. The comparison with a specialist's CM is also used to verify the number of valid and invalid propositions, as well as the presence or absence of concepts and relations considered critical.

Our research is innovative because it introduces the idea of using GAs and ontologies in the process of learning assessment.

5 Conclusions

In this article we describe the architecture of the CMTool environment, a tool developed to support the meaningful learning theory in classroom, focusing on learning assessment. As shown in section 4, most of the research about assessment using CMs compares the learner's CM to a reference CM, which appears to us to be unsatisfactory for constructivist learning.

Based on several experiments carried out in our education institution, we have understood that it was necessary to advance a little more in some directions to reach the objective of assessing precisely the learning types and cognitive styles used to learn. This is a difficult task as the learning process is personal (idiosyncratic) and the assessment based on the comparison between a learner's CM and a CM drawn by a specialist ends up reproducing the positivistic logic of behaviorism (Novak, 1998, p. 49).

Instead of using this approach, our proposal to assess a CM involves the use of ontologies and machine learning (through genetic algorithms – GAs). Ontologies can describe knowledge domains, but not the types of meaningful learning. On the other hand, the genealogic operators in a GA are capable of, based on an ontology, simulating the construction of the learning situations known as progressive differentiation and integrative reconciliation. Additionally, the assessor component, using a function stored in the ontology, can measure the semantic distance between CMs. This is important because the environment can present alternative learning situations to the student, as well as emphasize possible misconceptions. As future research, we are studying the use of the assessor in mining learning styles in collections of CMs. The objective is to identify dominant learning styles (progressive differentiation or integrative reconciliation). This is an important functionality, because it allows classifying students under cognitive preferences and offering personalized aid during the learning process.

Concerning the GA, two aspects need further consideration: the quantity of produced material and scalability. The dimension of the search space is a good measure of the quantity of material produced by the GA. It is proportional to the richness of the ontology and to the characteristics of CM_{ap}. In some situations, the search space may be greater than necessary, considering the learning task. Because of this, we are currently developing mechanisms capable of identifying and extracting significant samples of the search space. We have developed an artificial neural network (not described in this paper) capable of recognizing some semantic patterns in CMs (meronymy, holonymy, hipernymy, hyponymy, synonymy, and exemplification). The next step is to increase the quantity of recognizable patterns and incorporate this mechanism in CMTool. Concerning scalability, we have tested medium-sized ontologies (up to 100 binary relations) and verified that the GA converges quickly, if these relations are mapped up to 5 values. Once learners' CMs usually have 20 to 30 binary relations, the performance of the GA can be considered satisfactory.

Ausubel (2000) says that the cognitive structure is characterized by factors (or variables) that need to be enhanced for the improvement of education. Therefore, more research needs to be carried out to predict or measure stability, clarity and discriminability of the cognitive structure. Our results are a step towards this direction.

6 References

Araújo, A. M. T., Menezes, C. S., & Cury, D. (2002). *Apoio Automatizado à Avaliação da Aprendizagem Utilizando Mapas Conceituais*. In Proceedings of XIV Simpósio Brasileiro de Informática na Educação (SBIE – NCE-IM/UFRJ – 2003). Porto Alegre, RS: Sociedade Brasileira de Computação.

Ausubel, D. P. (2000). *The Acquisition and Retention of Knowledge.* New York: Kluwer Academic Publishers.

Cabral, A. R. Y. & Giraffa, L. M. M. (2002). *Avaliação de cursos WBT utilizando Mapas Conceituais*. In Proceedings of XIII Simpósio Brasileiro de Informática na Educação (SBIE – UNISINOS – 2002). Porto Alegre, RS: Sociedade Brasileira de Computação.

Cañas, A. J., Leake, D. B., & Wilson D. C. (1999). *Managing, Mapping and Manipulating Conceptual Knowledge*, AAAI Workshop Technical Report WS-99-10: Exploring the Synergies of Knowledge Management & Case-Based Reasoning, AAAI Press, Menlo Calif, (July 1999).

Chung, G. K. W. K., Baker, E. L. & Cheak, A. M. (2002). *Knowledge Mapper Authoring System Prototype: CSE Technical Report 575.* Los Angeles, CA: University of California.

Costa Jr, J. V., Rocha, F. E. L. & Favero, E. L. (2004). *Linking Phrases in Concept Maps: a Study on the Nature of Inclusivity*. In Proceedings: 1st International Conference on Concept Mapping, Navarra, Spain.(2004).

Cunha, M. J. S. & Fernandes, C. T. (2002). *AC3As-Web: Ambiente Cooperativo de Apoio à Avaliação de Aprendizagem Significativa na Web*. In Proceedings of XIII Simpósio Brasileiro de Informática na Educação (SBIE – UNISINOS – 2002). Porto Alegre, RS: Sociedade Brasileira de Computação.

Fischer, S. (2001). *Course and Exercise Sequencing Using Metadata in Adaptive Hypermedia Learning Systems*. ACM Journal of Educational Resources in Computing, 1(1), Spring 2001, Article #5, 21 pages.

McGriff, S. J. (2001). *Measuring Cognitive Structure: An Overview of Pathfinder Networks and Semantic Networks*. Pennsylvania State University

Novak, J. D. (1998). *Learning, creating, and using knowledge: Concept Maps as Facilitative Tools in Schools and Corporations.* Mahweh, NJ: Lawrence Erlbaum Associates.

Novak, J. D. & Gowin, D. B. (1999). *Aprender a Aprender*. Lisboa: Plátano Edições Técnicas.

Rocha, F. E. L. & Favero, E. L. (2004). *CMTool: A Supporting Tool for Conceptual Map Analysis*. In Proceedings of World Congress on Engineering and Technology Education 2004. Santos, Brazil, March, 2004.

Rocha, F.E.L., Vieira, R. V., Costa Jr, J. V., Favero, E. L. (2004). *Um Algoritmo Genético para Auxiliar na Avaliação da Aprendizagem Significativa por meio de Mapas Conceituais.* Submited to SBIE, Manaus, Brazil, 2004.

Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks Cole Publishing Company.

Turns, J., Atman, C. J., & Adams, R. (2000). *Concept Maps for Engineering Education: A Cognitively Motivated Tool Supporting Varied Assessment Functions*. In: IEEE Transactions on Education, 43(2), May, 2000.

Zbigniew, M. & Fogel, D. B. (2002). *How to Solve It: Modern Heuristics.* New York: Springer-Verlag.