

Development of Multi-Level Holistic Model in Integrated Learning

Catherine S. Lacsamana¹, Tomas U. Ganiron Jr² and Harold S. Taylor³

¹*Faculty of Science and Technology, University of Tartu, Estonia*

²*International Associations of Engineers (IAENG), Hongkong*

³*Department of Computer Engineering, Victoria University of Wellington, NZ*

¹*catherinelacsamana11@gmail.com, ²tomasuganironjr@gmail.com,*

³*haroldtaylor1113@yahoo.com*

Abstract

When providing for a group learning situation where each student was at their own machine, the need arose to combine individual and group learning models to provide an educational framework for a supportive learning environment. The framework had to be implementable on a (reductionist) computer network. The independence of behaviour and implementation was exploited to allow a holistic educational framework to be used when designing the learning environment. In a model, memory, functions, and goals are reducible, whereas the relationship between the purpose functions and the state manipulating functions are irreducible. Implementation was neatly achievable using the software engineering technology of object oriented technology to provide encapsulation for the reducible aspects while preserving holistic entities that can exploit the benefits of polymorphism.

Keywords: *Object oriented technology, holistic, encapsulation, polymorphism*

1. Introduction

Models vary from strongly directed frameworks for individual learning situations to those that only provide tools for social interactions. The need to combine both approaches arose for the authors when designing a learning environment for distant education students studying computer programming. Typically these students would work alone in a programming environment and then come together in a MOO or a forum. It was easy to find tools on the level of file transfer. However to find tools that directly supported learning at both levels were not easy to find because computer support for the transfer of the results of learning from one situation to the other requires an integrated model of learning. An integrated model of learning requires an integrated framework of learning. Certainly there are many individual tools based on a clear learning framework. As long ago as 1990, Anderson, Boyle, Corbett & Lewis provided an intelligent tutor based on an extensive learning framework [1]. Even so, most tools are fairly restricted in their approach. Indeed Siviter could still claim in 1996 that most tools were opportunistic and lacked any educational framework [2]. A typical support for concept association and abstraction provides tools, such as hypertext, but does not have a model of the actual concept to be learnt. Current typical learning frameworks of social learning environments of students are modified email, list servers, MOOs, forums, and rooms [3-7]. These provide the environment for the actors to interact using objects like blackboards, libraries, and noticeboards. They usually avoid modeling the group learning as such, and certainly they do not provide a model as to how individual learning can be integrated into the group learning. Without such a model, it is difficult to hold group concepts. It is even more difficult to transfer concepts between the two environments. In designing a learning

environment for distant education students, the author had to address the problem of transferring evidence of concepts between each student's own study environments and a net-based group learning situation, and vice versa. It was done using a two level model based on object oriented technology. One of the central issues of object oriented technology is the encapsulation of sub-entities. Encapsulation separates the behaviour of an entity from the implementation of that behaviour with the result that the behaviour can be treated in a holistic educational framework on a model implemented using reductionist computer technology.

2. Issues in Combining the Individual and Group Models

There are three aspects to a model - state, functions, and goals. State is stored in the memory of the model. The ontology of a model is the set of valid different states. A function is a well defined rule for a change of state. A particular memory could be either atomic or divisible into smaller parts. If the memory is atomic then the ontology is set for that item. If the memory can be divisible into smaller parts, then a function for changing the state of the memory can also be divisible into smaller steps. If there are fewer types of sub-memory parts than there are types of memory, then the ontology of the model has been reduced. Reductionism includes the belief that all models of reality can be reduced to a few entities, and it is these entities that provide an explanation for the behaviour in the model. Rationalism does not admit that higher levels can have a goal [8,9]. Holistic descriptions also include a goal or purpose [10,11]. A goal is a desired state expressed in terms of the current state, and the available functions. When the model is subdivided, the goal will also be subdivided, so that it can be expressed in terms of the sub-model. The significance of a change of state on the lower level can only be understood by the grouping of sub-goals together into the higher level goal. At the individual level, the constructivist theory of concept formation indicates that a student forms new concepts by reorganizing old concepts to match incoming perceptual data when that data is sufficiently discordant with the old concepts as to require significant conceptual load [12]. The concepts can be modeled as a concept map. Reorganization of concepts will then be represented by map transformations that comply with constructivism [13]. Groups also learn when undertaking a project. Simon asserted that language allowed a group to undertake projects that were bigger than could be handled by a single person [14]. In this case, the tutorial group has a group goal, namely to develop professionally acceptable problem-solving concepts as indicated by the lecturer, textbooks, and tutor. The mechanisms in a group's learning are similar to the mechanisms in an individual's learning [15]. This means that there are group concepts that can also be represented in a concept map. Constructivism is a holistic theory in that it specifies a goal. For the individual, the goal is the reduction of individual cognitive load; for the group, the learning goal is harmonization of concepts so that an external goal can be achieved. A two level holistic model of cognition was required. In this section, we discuss some of the features of multi-level holistic models and the techniques that object oriented technology provide for their implementation.

3. Relationships between Levels of a Model

3.1. Separating Levels

One common error in multi-level models is to mix issues from different levels, because each of the aspects (memory, function, and purpose) may have validity at one level but not at another level [16]. In this case, the tutorial group has a group goal, namely to develop professionally acceptable problem solving concepts as indicated by the lecturer,

textbooks, and tutor. A professionally acceptable problem solution is the evidence of professionally acceptable concepts in the group's cognitive space. Operations are social negotiations such as "a less-powerful member of the group adjusts their understanding to be more consistent with the most powerful people in a group (in this case the lecturer)". This goal is only makes sense at the group level. For each student, the overall goal can be divided into a sub-goal for each student that each one adjusts their concept-map to minimize the cognitive work required to understand the group's problem-solution. To achieve the goal, the student modifies their concept map, using the functions of constructivism, namely progressive differentiation, integrative reconciliation and super-ordinate learning. The memory of the student model is the concept map. If there were to fail to change the level of purpose when changing the model level, then find a complete social negotiation mechanism, as opposed to a concept changing mechanism, in the student's mind. Conversely, the efficiency of the individual cannot be imported into the social negotiation mechanism without losing the essential power relationships of the negotiation relationships. Clearly, in a multi-level model, a level may provide a place for the storage of data, functions, and goal. When moving down levels, to subdivide both the goal of an explanation as well as the functions and memory. The goal of a model must always be expressed in the behaviour of storage units valid at that level.

3.2. Atomic Entities for the Individual

It is clear that each level relies on the level below to supply sub-memory and sub-function, its own level to integrate them into memory and function at this level and the level above to supply purpose. An entity moving from a lower level to a higher level is atomic at the higher level. Often constructivists refer to creations of the individual's mind as concepts and input from the environment as data. Others recognize that all data that the mind perceives consciously has already been processed, so refer to new input as concepts. These entities are atomic concepts. An atomic concept is one that comes from the sub-conscious processing of perceptions [17]. To accommodate this implicit knowledge, Searle included in a model of the mind a set of propositions that form the "Background" [18]. The Background is the set of non-representational mental capacities that allow all representing to take place [18]. The Background provides a mechanism whereby atomic concepts can originate. Others hold that propositions are too ordered to explain the Background processing [19]. Indeed, Lakoff argued that the basic level reflects the bodily nature of people supporting some possible processing but denying other processing [20]. If an individual could be modeled when they first had experience but no concepts then their concept map would consist of nodes representing atomic concepts. Concepts are then developed to reduce the cognitive load of accepting more atomic concepts.

3.3. Atomic Entities for the Individual

If the members of the group came together with no previous common experience, the group concept map would consist of a single node representing the group with a sub-node for each member. (Encapsulated in each sub-node would be the student's own concept map.) As each tutorial group interchanges ideas, the group builds up group-concepts that represent an amalgam of individual contributions. These group-concepts may, for example, be used to develop a program that is different from the program that any member of the group could develop. Each concept that is discussed, and a common meaning agreed to, is an independent node in the group's concept map. If all participants agreed on the exactly the same meaning, then the node would be a single node in the concept map. More likely, each person has a consistent, but individualistic, concept. This concept is a refinement of the common concept. The refinement part of the concept remains in the individual's node. This approach allows the group model to be studied as a phenomenon in its own right. It

has group goals, social functions to achieve those goals, and the group memory. Group memory includes group resources, like whiteboards, that are independent of individuals. Each of the memories is modeled by itself.

4. The Benefit of Encapsulation

It has been noted that different levels in the models need to be encapsulated. A particularly useful technology to implement this is the object oriented technology. The three bases of object oriented technology are encapsulation (for information hiding), classification (of abstract data types), and polymorphism through inheritance as understood in object oriented technology [21]. A model that uses encapsulation is an easy way to model concepts. It has been suggested that the human mind is a symbol manipulator [14], where a symbol is a (discrete) physical pattern that can be used to create structures. This is implicitly recognized throughout this paper by use of the discrete circles of a concept map to represent concepts. Henderson-Sellers note that information hiding requires that an encapsulated entity does not allow other objects to see its internal structure [21]. This separates the holistic behaviour of an object from its internal reductionist representation so is convenient. (This separation is natural for concepts and students but is less natural to observe at the tutor group level as often confuse in personal participation within the group with modeling of the group.) An abstract data type, or user-defined data-type, is the description of an object. The description uses other object descriptions. The sub-objects in a particular object have their own boundaries so objects are a set of nested encapsulations. Such a model is reductionist in that the parts are separable. The behaviour of an object describes in ways that use the internal structure. This can describe an object's behaviour solely in holistic terms, in terms of its methods (functions). Such an abstract behaviour is nearly orthogonal to the internal implementation [14]. An object that behaves in a particular way, can have many implementations. Classification is the grouping of software objects into hierarchically organized categories. Common behaviour in two classes can be placed in a more abstract class higher in the tree and inherited by the sub-classes. In this case, a tutorial group object and a single student object have common behaviour, so the class descriptions can descend from a single abstract "Learning Situation" class. Because the memory structure of the two classes of objects are very different, the behaviour of the two classes is polymorphic, Henderson-Seller's third base. (Polymorphism is the ability of classes to share abstract behaviour, without sharing implementation.) This is shown in Figure 1.

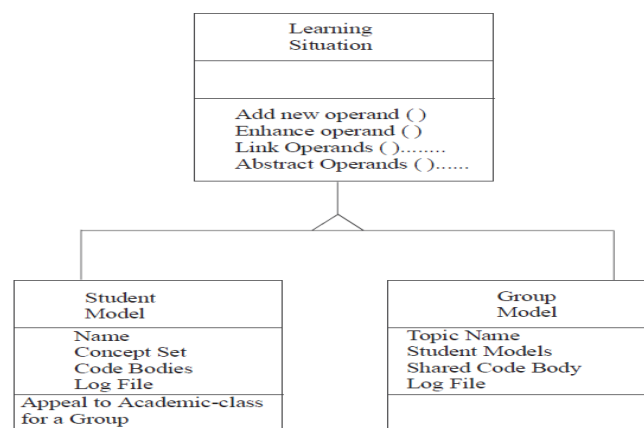


Figure 1. Class Diagram with Abstract Learning Situation Specialized into Concrete Models

It can now be asserted that the separation of the internal reductionist description and the external holistic behaviour allows a start in combining the two conceptual frameworks by providing a mechanism for accommodating the two conceptual frameworks in one model.

5. Dividing Goals - non-determinism, Determinism, and Reification

Encapsulation helps to contain the philosophical problem of goal setting. It is possible that a model is created with no goal. For example, Dawkins claims that all direction comes from the genes in biology [9, 22]. As a result, Dawkins denies that animals have any purpose other than to be gene carriers. This has two implications for a model. Firstly, genes have to get their “purpose” from an internal progression generator such as a random number generator. Secondly, the structure of animals is just a convenience - the model is essentially “flat”. These implications can lead to unmanageable models very quickly if there are many levels. If an extra protein lower level is added to the biology model, then the genes cannot have any purpose other than to be protein carriers. The internal progression generator cannot be at the gene level, but must be at the protein level. The flatness is transitive, so animals have no purpose other than to be protein carriers. This argument can be repeated for each level. Ultimately the internal progression generator has to be located at the lowest level modeled e.g. at quark level. To argue that animals exist, simply so that quarks can show quark behaviour seems somewhat extreme. Equally, a model created with all the goals at the top level can seem extreme. If all the purpose of a student is dictated by a higher group level, then the group is totalitarian, and is only using the individual student as a convenient memory location. Again, the group is part of a class so the class must be authoritarian. The same argument can be repeated for higher and higher levels to some abstract life force. To argue that students exist, simply so that the university has a convenient place to store information also seems somewhat extreme. To avoid either extreme, assume that every necessary level of modeling can both have purpose transmitted from a higher level and have purpose that is outside that higher level control. That is, each encapsulation is reified into an entity that may contribute some purpose that is independent of the group’s purpose. In computer modeling terms, reification requires not only encapsulation but also the encapsulated entity has to be able to react independently so it needs to be located on its own logical thread. Note that reification does not forbid reductionism of the state of the entity, or reductionism of the functions of the entity. However, some functions will provide purpose or goals. It is the relationship between the purpose functions and the state manipulating functions that is irreducible. Consider the case of reducing the individual student mind (consciousness) to a set of concepts. On the one hand, the mind is not reducible for educational philosophical reasons [22, 23]. This irreducibility of consciousness is important because it loses the subjective feeling, and our integrity of action, if we subdivide the functionality of consciousness [19, 24]. On the other hand, the mind can be subdivided into concepts because reification does not mean that we cannot subdivide the mind to find how parts contribute to the functionality of the mind, but just that the subdivision inevitably loses the totality of thinking. When modeling a student mind in a group situation, it is convenient to encapsulate the model of student memory and its functions, it is convenient to encapsulate the functions of control, but we must add a separate thread for the mind model if we wish to provide a model of consciousness. A simple way to provide each student model with its own encapsulation and thread is to locate each model on its own computer in a network of computers. It can now be asserted that the reification separates the internal reductionist description and the external holistic behaviour of an entity with purpose allows the two conceptual frameworks to be combined in one model.

6. A model of Cognitive Space

6.1. Concepts

Primitive concepts form the atomic level of the model of the student's concept map. They have a name and content. The name and the content are often the same. Primitive concepts also know other concepts that are linked to them. Derived concepts are concepts that represent a group of concepts linked using constructivist functions. The evidence that a student has a particular concept of a problem domain in their mind will be in the vocabulary of names that they use. In this case, the evidence that a student has a particular programming concept in their mind will be the use of the names of that concept in a program.

6.2. A Model for an Individual

The cognitive space of a student was modeled with memory being a concept map, and functions being those allowed by constructivism [13, 25]. The model assumed that the memory of a student could be divided into concepts together with linkages between the concepts. Each link was either an is-a link, or an is-a-part-of link, or a relational link. The model provided functions such as "relationLink(concept1, concept2, relationName)". The goal of the cognitive space of the student is to minimize cognitive load. For example, when a student is constantly doing cognitive work to relate two independent concepts, cognitive load can be reduced by introducing a relational link between the two concepts. The model called for guidance from a concept map of the discipline that was based on an analysis of the textbooks (approximately 1100 concepts). The model was augmented with each concept also holding the evidence for that concept. (For example, concrete loop structures in a program are evidences that the student holds the loop as a concept.) The evidence of program is shown in the windows for code, data input and output windows, current status of variables, an error window, together with tutor help window(s). The evidence of the change of conceptual structure is recorded in the programming environment's log.

6.3. A Model for a Tutorial Group

A tutorial group is a tutor and a selection of a small number of students. It is expected that the tutor's concept map is consistent with a concept map of the discipline! Let the students be reasonably matched in that all students are currently working on the same part of their concept map. Further, let the students be having difficulties in modifying their concept maps to be consistent with the concept map of the discipline. The students can contribute their programs as evidence of each of their current concept maps. The tutor could provide a program that is consistent with the concept map of the discipline. Thagard asserted that both groups and individuals do hold concepts using a method that is primarily structured via kind-hierarchies (abstraction) and part-hierarchies (aggregation) [15]. So a model of the group memory is a concept map of the group as a whole. (In fact, the common node would be redundantly stored in each person's memory, together with their individualistic concepts.) Thagard, a cognitive scientist, was concerned with the development of such a group concept map. (In his case, the group concept map represented a scientific field.) Thagard's functionality of a group concept map is similar to that of a constructivist's functionality of an individual's concept map (see Table 1).

Table 1. Similarities between Modeling Levels

Characteristics	Individual Mind	Group Programming
Operands,	Concept Individual's Concept Set	Group Concept
Language Style	Monologue	Dialogue
Removing operands	Obliterate Subsumption	Forgetting, Absorption
Adding new operands	From Background	From Individual Mind
Enhancing operands	Progressive Differentiation	Differentiation
Cross-Linking Operands	Integrative Reconciliation	
Aggregating Operands	Combinatorial Learning	Revisionary Organization
Abstracting Operands	Super Ordinate Learning	Abstraction
Re-arranging map		Hierarchy Redefinition

7. References Exploiting Similarity - A benefit of encapsulation

7.1. The Student Interface

So far, it has been shown that encapsulation allows the separation of a holistic behaviour by hiding the reductionist implementation. Since the behaviour of an object of the group learning model class is very similar to the behaviour of an object of the individual learning model class, the two classes can be made polymorphic - Henderson-Sellers's third base of object oriented technology. Indeed, the two classes can descend from one abstract object as in Figure 1. The advantage from a programming point of view of developing the abstract class is that client objects can use the abstract class interface without knowing with which class they are actually interfacing. One such client object is the student interface, so the advantage from the student's point of view is a uniformity of interface between the individual environment and the group environment - as they are in a real classroom. Private learning and public learning are not disjoint activities. At the individual level, the interface has to provide windows to show concept discussions from the help files and for different aspects of the programming. At the group level, there will be concept discussions with the (human) tutor, and windows to represent different aspects of programs e.g. each student should have their own window to display their program and there will need to be a window for the mutually agreed program. At the individual level, linking is currently done implicitly or by the programmer environment. At the group level, students will discuss each other's code (negotiate), linking implicitly, as annotations, or perhaps more graphically. Objective code will be that agreed to and placed in a common window. Providing a single interface raises the possibility that the interface can be written once and then can provide the student with a more uniform interface in two different activities.

7.2. Generality

There are good arguments for a variety of classroom arrangements. For example, even within the tutorial group students might be encouraged to work in small groups. When the lecturer is forming such a group, they may wish to collect both individual students and other small groups. The life of these groups may be independent of the life of the original group. Another possibility is that not all members of the group are available at once. Provide a mechanism for easy addition of new members, as they become available. These considerations suggest that we may wish to join students together, a student with an already existing tutorial group, and perhaps tutorial group with tutorial group. The uniformity of the behaviour of the two learning situations makes any of these combinations simple to implement.

8. Conclusion

Complete encapsulation, as provided by object oriented technology, provides a mechanism for implementing a holistic learning environment on a reductionist computer. The encapsulation allows the behaviour of the learning model to be separated from the implementation of the memory and functions of the learning model. A two level model was used - one level for the individual workspace and one level for the group work space. For the individual a concept map was used, where concepts could be atomic concepts arising from the background or they could be concepts derived using constructivist transformations. For the group a concept map was also used, where concepts came from individual students or they were concepts derived using social constructivist transformations. Care had to be taken not to break the encapsulation when modeling. This is easy to do when the modeler can identify with one level more than another. Encapsulation can be taken further to reify the learning environment object. Reification allows the creation of additional goals at a level other than the extreme highest, or lowest, level. The similarity of the behaviour at the two levels of the model can be exploited if object oriented technology is used to implement the learning environments as the same code can be re-used in each model. In particular, the interfaces can be made uniform, which makes for a more usable environment, and secondly, the range of rearrangements of class structure is considerably widened. There has been a tendency for application disciplines to become more reductionist in their modeling as computers are used more often. This study showed that that is not always necessary.

References

- [1] J. R. Anderson, C. F. Boyle, A. T. Corbett and M. W. Lewis, "Cognitive Modeling and Intelligent Tutoring", W. J. Clancey & E. Soloway (Eds.), MIT Press, (2009).
- [2] T. U. Ganiron Jr, "Planning and Execution of Holistic Model Using Object Oriented Technology", Proceedings of the 3rd International Conference The Future Education, Florence, Italy, (2013) June 7.
- [3] Z. Fadeeva and Y. Mochizuki, "Higher Education for Today and Tomorrow: University Appraisal for Diversity, Innovation and Change Towards Sustainable Development", Sustainability Science, vol. 5, no. 2, (2010), pp. 249-256.
- [4] M. Čudanov, G. Savoju and O. Jaško, "Usage of Technology Enhanced Learning Tools and Organizational Change Perception", Computer Science and Information Systems, vol. 9, no. 1, (2012), pp 285-302.
- [5] T. Ucol-Ganiron Jr. and A. Alaboodi, "Cultural Learning Environment in Structural Engineering Courses of Architecture and Civil Engineering Students in Qassim University", Procedia-Social and Behavioral Sciences, vol. 102, (2013), pp. 300-310.
- [6] T. Ucol-Ganiron Jr, "Student Competencies in Structural Engineering: Modelling Cultural Environment in Qassim University", Journal of Proceedings of the 40th Annual Conference of the European Society for Engineering Education, Thessaloniki, Greece, (2012) September 23-26.
- [7] T. Ucol-Ganiron Jr, "Cultural Learning Environment and Perception of Structural Engineering Classes in Qassim University", WSEAS Transactions on Advances in Engineering Education Journal, vol. 9, no. 3, (2012), pp. 84-93.
- [8] T. U. Ganiron Jr, "The Effect of Study Group Activity Guide in Expository Reading and Writing Course at the College of Architecture in Qassim University", International Journal of Education and Learning, vol. 3, no. 1, (2014) March, pp. 23-34.
- [9] T. U. Ganiron Jr, "Teaching Styles of using English Drama in Critical Thinking Course for Architecture Students in Qassim University", International Journal of Education and Learning, vol. 3, no. 2, (2014) September.
- [10] N. Mlitwa and J. Belle, "A Proposed Interpretivist Framework to Research the Adoption of Learning Management Systems in Universities", Communications of the IBIMA, (2010).
- [11] D. Tesoro-Martinez, T. U. Ganiron Jr and H. S. Taylor, "Multimedia Tools for Teaching Basic Electronics", International Journal of Education and Learning, vol. 3, no. 2, (2014) September.
- [12] D. T. Martinez, T. Ucol-Ganiron Jr and C. S. Lacsamana, "Development of Hardware Interfacing System for Visual C++", International Journal of Advances in Applied Sciences, vol. 2, no. 4, (2013) December, pp. 1001-1005.
- [13] M. Clarke and M. Killeavy, "Charting Teacher Education Policy in the Republic of Ireland with Particular Reference to the Impact of Economic Recession", Educational Research, vol. 54, no. 2, (2012), pp. 125-136.

- [14] C. S. Lacsamana, T. U. Ganiron Jr and H. S. Taylor, "Developing Low Cost Laboratory Apparatus for Hardware Interfacing System", International Journal of u- and e- Service, Science and Technology, vol 7, no. 2, (2014) April, pp. 113-126.
- [15] T. Ucol-Ganiron Jr, "Structural Engineers Career Success", International Journal of Innovation, Management and Technology, vol.3, no. 4, (2012) August, pp. 321-326.
- [16] T. Ucol-Ganiron Jr, "The Additive Value of Psychological Capital in Predicting Structural Project Success and Life Satisfaction of Structural Engineers", International Journal of Social Science and Humanity, vol. 2, no. 4, (2012) July, pp. 291-294.
- [17] T. Ucol-Ganiron Jr and T. Malvecino-Ganiron, "Managing Career Success of Geodetic Engineers", International Journal of Education and Learning, vol. 2, no. 1, (2013) March, pp. 13-24.
- [18] T. Ganiron Jr and D. Tesoro-Martinez, "Investigation of the Predictors of Civil Engineer Career Success", Journal Proceedings of the 41st SEFI Conference, Leuven, Belgium, (2013) September 16-20.
- [19] T. Ganiron Jr, T. Ganiron III and N. Ucol-Ganiron, "Modeling the Level of Objective & Subjective Career Success of Civil Engineers towards Developing a Career Planning Program", Journal Proceedings of the 2012 International Conference on Information and Knowledge Management, vol. 45, no. 1, (2012), pp. 36-41.
- [20] T. Ucol-Ganiron Jr and T. Malvecino Ganiron, "Social Capital on Civil Engineer Career Success", International Journal of Innovation, Management and Technology, vol.3, no. 6, (2012) December, pp. 718-724.
- [21] T. U. Ganiron Jr, "Applications of Accelerated Learning in Teaching Environmental Control System in Qassim University", International Journal of Education and Learning, vol. 2, no. 2, (2013), pp. 27-38.
- [22] T. Ganiron Jr, "Accelerated Learning Techniques: Teaching Critical Thinking in Qassim University", Journal of Proceedings of the 41st Annual Conference of the European Society for Engineering Education, Leuven, Belgium, (2013) September 16.
- [23] P. Känkänen, Päivi and M. Bardy, "Life Stories and Arts in Child Welfare: Enriching Communication", Nordic Social Work Research, (2013), pp. 1-15.
- [24] T. Ucol-Ganiron Jr, "Web-Enhanced Project Management Course", International Journal of u- and e-Service, Science and Technology, vol. 6, no. 1, (2013), pp. 49-59.
- [25] T. Ucol-Ganiron Jr, A. Alaboodi and A. Eltoumi, "Designing and Developing a Web Enhanced Project Management Course for Engineering Graduating Students", Journal of Proceedings of the International Forum on Engineering Education, University of Sharjah, United Arab Emirates, (2010) November 23-25.

Authors



Catherine S. Lacsamana, was born in San Nicolas, Ilocos Norte. She received both BS and MS Accountancy from De La Salle University, in Manila. Ms. Lacsamana has been pursuing research in financial accounting and corporate governance. She has published over 25 referred papers in relevant journals and conferences, which majority focus on the areas of social, environmental and ethical accounting and reporting, and corporate and social responsibility.



Tomas U. Ganiron Jr., This author obtained his Doctor of Philosophy in Construction Management at Adamson University (Philippines) in 2006, and subsequently earned his Master of Civil Engineering major in Highway and Transportation Engineering at Dela Salle University-Manila (Philippines) in 1997 and received Bachelor of Science in Civil Engineering major in Structural Engineering at University of the East (Philippines) in 1990. He is a registered Civil Engineer in the Philippines and Professional Engineer in New Zealand. He has received the Outstanding Civil Engineer in the field of Education given by the Philippine Media Association Inc. (1996), ASTM Award CA Hogentogler (2008) by IPENZ in New Zealand and Outstanding Researcher (2013) in Qassim University, Buraidah City.



Harold S. Taylor, received the BS Civil Engineering degree at University Canterbury in 1983 and earned his MS (1985) and PhD (1987) degrees in Civil Engineering at University of Auckland. He has published several research papers in national and international journals in the area of hydrology, water resources engineering, waste treatment plant and river engineering. He has also authored books viz, Ocean Engineering, River and Sediments, Water Resources and Engineering, and Fluid Mechanics Students' Handbook.