RESEARCH ARTICLE OPEN ACCESS

Re-Engineering Learning Objects for Re-Purposing

Al-Khanjari, Z. A and Kutti, N. S.
Department of Computer Science, College of Science, Sultan Qaboos University

ABSTRACT
Existing Learning Object (LO) definitions and their interpretations seem to project a view of LO with somewhat less flexible in the scope of LO reusability.In solving this problem the researchers try to interpret those definitions with an added property i.e. 'LO repurposing'. LO repurposing refers to the ability of reusing the LO in different perspectives and contexts. Adding new property needs a suitable Software Engineering methodology to be applied to properly inject the required strategies toward the development of the best and quality solutions. Our investigation in finding a solution for this problem is by shaping the LOssimilar to software objects that can not only be reused, but also repurposed in various learning contexts. In our paper, we propose to apply an object-oriented framework to develop the class-based LO model with an evolving nature of LOs. The class-based LO model allows LOs derivable to the degree ofsufficient level of repurposing for any learning context.
Keywords-O-O LO model, LO class, domain-based LO hierarchy, data object, LO repository, LO reusability.

I. INTRODUCTION

Two basic criteria, first the cost of developing good learning resources and secondly launching pedagogically effective learning contents prompted the researchers to look for learning resources that are reusable, updatable, self-content and adaptable to various learning contexts.

There are several LO models under different names developed based on the basis of usability and reusability.

The Netg, a Computer-based Training (CBT) vendor [1] implemented a multiple component LO model. It is basically consisting of three structural components: a learning objective, an instruction unit and an assessment unit. An important feature of this model is that each component is independent of its meaning and structure and can be stand-alone. LO may be reused in a different learning context, but it is presenting a passive or simply reading element.

Cisco Systems defined their learning object model as "a granular, reusable chunk of information that is media independent" called Reusable Learning Objects (RLOs). Cisco adopted a strategy for designing the instructional component of RLO in a modular fashion. They used Reusable Information Objects (RIOs) as building elements in constructing the instructional component. The RIOs are managed in a repository as the learning resources at the lowest level [2]. A RLO can have five to nine RIOs to achieve a unique learning objective. To support learning experience, other components such as an overview, summary and assessment are added. All these components form a Reusable Learning Object (RLOs). The Reusable Information Object is in turn composed of three items: content items, practice items, and assessment items. In their standard both RLOs and RIOs are structured, the former represents a content model based on a learning objective and the latter represents simply an information or knowledge element. The model facilitates to construct a simple, but at the same time pedagogically rich RLO serving a single learning objective. This may imply that RLOs are closely tied with a learning context, thus reducing the reusability of learning objects in other contexts.

It is interesting to note that the RLO structure allows to extend the existing RLOs by adding new RIOs or replacing with alternate RIOs as needed. Thus the architecture can be optionally used to support repurposing, thus increasing the scope of reusability. The CISCO's RLO model is definitely a step closer towards software objects. However, there is no organized framework to repurpose the existing LOs.

In 2000,Hodgins[3]coined the name "Learning Object" after discovering the possibility of creating reusable learning resources based on the LEGO metaphor. The attractive feature in his finding was reusability because of its cost saving nature. An interesting observation by Wiley [4] and others showed that reusability feature of LO model resembles the basic feature of most of the modern software components developed under the modular and object-oriented principles.

In 2001,Wiley [5] opposed the idea of using a non-digital resource as LO. He redefined an LO as any digital resource that can be reused to support learning. An LO in this definition includes any resource irrespective of its size that can be delivered across the network on demand. This aspect of the definition strongly emphasizes a need for a true Object-Oriented LO model. The abstract nature of these definitions helps indirectly the learning content developers to develop their own LO models according to their requirements, but at the same time with the reusability feature.

In 2002, Boyle [6]in his earlier work described his LO model to possess both reusability and repurposing. He configured his LO model using ‘Compound Object’ structure mainly to serve repurposing. The design has also adopted ‘cohesiveness’ and ‘decoupling’ to enhance reusability characteristic of LOs. The resulting model allowed adding more elements or objects to form pedagogically a suitable learning object for a targeted application. The model indirectly supports a sort of inheritance similar to software objects. The cohesion property of this model specifies that the LO should be atomic and supporting only one objective or goal. The de-coupling property specifies that the LOs do not depend on other LOs and thus allow reusability without any constraint. This compound object model is also a step forward in shaping LOs in line with true software objects.

Anotherwell-known LO definition by the IEEE Learning Technology Standards Committee (LTSC) Learning Object Metadata (applications is [7]:

“Learning Objects are broadly defined as any entity, digital or non-digital, which can be used, re-used or referenced during learning supported by technology and supported by learning systems such as computer-based training systems, interactive learning environments, intelligent computer-aided instruction systems, distance learning systems, and collaborative learning environments. Examples of Learning Objects include multimedia content, instructional content, learning objectives, instructional software and software tools, and persons, organizations, or events referenced during technology supported learning”.

This definition is more elaborative, but is also abstract. Towards e-Learning, it does not specify:‘LO structural significance’, ‘LO granularity’ and ‘LO effective organization'. According to the above definition any information object in its original form, even in a primitive formis accepted as an LO. The purpose of this definition is to guide the process of LO implementation to achieve a common criterioni.e. reducing the cost of learning contents. The definition explicitly mentions about the reusability nature of LO, LO-contents in various styles, learning goals through objectives and a need for an agent to launch the LOs in an organized sequence for an educational value. One of the drawbacks with this definition is that it assumes an LO can be reused in any learning context without any modification. However, this definition has more or less sufficient information to realize the need for a suitable LO model according to the given pedagogical requirements. Here, pedagogical requirements include both the scope of the final product and instructional strategy. Ultimately developing a feasible LO-model and launching the same successfully for a given scope and requirement specification needs a suitable software engineering methodology.

All these objects models have one principle in common. That is, the reusability feature of the LOs can simplify and reduce the time and cost of the process of building course contents. The LOs are also extended with varying structural properties for controlling the relationship between granularity and the reusability. However, it is not clear all those objects models mentioned above are truly supporting all features of Object-Oriented concept.

Some LO models presented previously try to relate the LOs with the software objects because of their reusable feature. In general, LO models including [8] may be interpreted in one hand that reusability can be achieved by associating only the cohesion and de-coupling characteristics with the LOs. Such LOs would tend to exhibit more of abstract nature rather than detailed one. The higher level aggregations such as lectures, chapters and courses based on this LO model would eventually result at abstract level products rather than being effective for an intensive pedagogical training. On the other hand, LOs having cohesive and decoupling properties made perfectly to a particular learning context may not be suitable at all to reuse effectively in different learning contexts. Hence, the LO design process needs a review over the existing definition to adopt as much as possible a reconfigurable data model to accomplish both reusability and repurposing.

Ed Morries [9]developed the ‘Learning Object Class (or LOC)’ as a template from which individualized LOs those can be inherited dynamically during learning session. His work is basically an extension of Boyle’s ‘Compound Object LO’[6]. He has induced the idea of inheritance into LO development. Both models aim to achieve a suitable repurposing mechanism which in turn helps to improve the effective reusability of an LO.

Ray Jones and Tom Boyle in their work [10] reused the patterns derived from the existing LOs in the design of new LOs. A pattern is also a class like abstraction similar to the Learning Object Class template developed by Ed Morris [9]. Patterns can be used to inherit/ implement pedagogically suitable LOs for new learning contexts. The patterns basically represent abstract problem solutions. Each pattern can be found useful to implement and/ or inherit suitable LOs to a particular learning context. The model is closer to software objects than the previous models discussed above.

In fact the research and the framework proposed by [6], [9] and [10] have dramatically changed the basic definition to reflect the repurposing property very significantly.

The need for software engineering discipline for LO development can also be observed from the models described by standards organization like the Department of Defense Advanced Distributed Learning (DoD-ADL) that came with a standard LO model called Sharable Content Object (SCO) [11]. It is also called SCORM reference model. Assets are used as building elements of instructional component in SCO. The SCORM standard clearly defines its assets as web-displayable knowledge elements. For example, assets can be graphics, HTML-documents, XML-documents, Flash files, audio/video, etc.

With the reusability criteria SCORM [12] uses the name "Sharable Content Object". SCORM gives more importance on the granularity of its Learning Object. According to SCORM philosophy the size of a SCO is preferred to be a small for easy maintenance and updating. At the same time SCORM allows the granularity of SCO even at different aggregation level as long as it is based on a single learning objective at that level. Moreover, the granularity of SCO depends upon a collection of all related materials (i.e. assets) to fully support the objective for which it is created.It seems that pedagogically rich SCOs can be constructedby collecting as many assets as needed. Hence, the granularity of SCOs indicates indirectly the level of richness with the contents. But, the reusability of SCOs is limited because repurposing is not facilitated by this model.

In 2009, Dimitriadis and colleagues [13]discussed the design and repurposing of the Open Educational Resources (OER). They explained how to make the inherent design of OER, more explicit to make them more understandable and reusable. Also, they stated that using a set of simple patterns of OER, increases the repurposing and ability to use them in a different context.

In 2010, Al-Khanjari and colleagues [14] used the idea of simple LO with a single objective towards the reusabilityand repurposing in different e-Learning contexts. Idrosa and colleagues [15]strongly recognized that the LO with a single objective can be used and repurposed in a multiple e-Learning context.

In 2011, Pegler [16]in his PhD thesis studied the importance and implication of reuse of digital resources. His study provided recommendations of how to prepare and utilize digital online resources. Sampson and Zervas [17]proposed a workflow for LO lifecycle and reuse. Then, they used the workflow to define metrics to measure the cost effectiveness of LOs reuse. Kaldoudi and colleagues [18]discussed educational content and their different repurposing context in medical education. They proposed a novel approach to content repurposing using Web 2.0 social networking of learning resources.

In 2012, Holz-Clause and Guntuku [19] proposed an RLOs (Reusable Learning Objects) approach for rapid development of e-Learning, which should be reusable, interoperable, durable and accessible. They showed how these resources can be reused and repurposed in different e-Learning contexts.

In 2013, Rufer and Adams [20]in their case study demonstrated how reusable and repurposable LO can enhance learners' knowledge in e-Learning environment. Minovic and colleagues [21]proposed a solution depending on educational games for successful reusing and repurposing of LO in multiple e-Learning contexts.

In 2014, Sultan and colleagues [22]stated that LO reusability helped in saving time and effort for educators and improving quality of e-Learning digital resources. They proposed LOREM (Learning Objects Reusability Effectiveness Metric) approach to evaluate LO based on a group of aspects which measure their repurposing level in different contexts. Keeping in mind reusing and repurposing of e-Learning digital resources, Piedra and colleagues[23] focused on openness of the educational resources. Their idea was to create and keep the digital resources as open licensed reuse to facilitate reusing the material in multiple contexts, which achieves repurposing.

This paper presents our on-going research in developing an Object-Oriented LO framework for facilitating with repurposing which enhances the reusability of LOs in a wide variety of learning contexts. Our framework uses the 'Compound Object' concept [6], 'Learning Object Class (LOC)' [9] and 'Reuse of Learning Object Patterns' concept [10] to develop a class-based LO hierarchical model for achieving repurposing. We apply 'Reuse of Design Classes' of the Object-Oriented Programming and Design principles in developing the class-based LO hierarchysupported by inheritance, implementation and over-riding principles to enable repurposing. The LO development follows Object-Oriented Software Engineering (OOSE) approach [24] and[25]. The instructional designers can use the three OOSE principles in creating and deriving LOs for various applications. The features of our framework allow creating class templates, class implementation from templates, class inheritance and class instantiations. The class inheritance allows the reusing of existing features of parent class either directly or after required mutation and adding new features for repurposing. We can identify some major advantages of this framework: (1) cost and time saving due to effectively reusable and repurposing nature of LOs and (2) designing LO-class templates appropriate to different pedagogical methods and (3) LO searching made simple in the LO hierarchical model. This model is discussed further in Section III. The rest of the paper is organized as follows: Section II explains the structure of basic LO. Section III discusses the engineering approach for LO structuring. Section IV discusses LO class hierarchy. Section V explains the designing process of the LO class hierarchy. Section VI provides concluding remarks of the work.

II. STRUCTURE OF BASIC LO

The data model should possess features such as cohesiveness, decoupling and supporting a pedagogical method. In addition the data model needs to incorporate metadata for searching purpose. Such LO has basically three kinds of information components, namely attributes related to metadata, and attributes related to learning contents and attributes related evaluation. In order to satisfy both reusability and repurposingpropertiesthe object-oriented LO model is identified as a suitable framework which can derive application specific LOs from a root LO. This requires modularity in composing the attributes. The content and evaluation attributes are sequences of content elements (called data objects)designed for a preferred pedagogical method or style. Figure 1 shows the LO structure in a schematic diagram form.

LO-Class: Meta Data Attributes (Indexing Data and Learning Objective)
LO-Class: Content Attributes (Multimedia Learning Data Elements)
LO-Class: Evaluation Attributes

Figure 1: A LO Class

III. ENGINEERING APPROACH FOR LO STRUCTURING

(a) General Description	(b) Power Point Object	(c) Quiz Object
(Data Object) CInteger Array: Meta Data	(Data Object) CRealArray: Meta Data	(Data Object) CRealArrayQuiz: Meta Data
General description (Text data)	Definition (Power Point data)	Quiz (Quiz Tool)

Figure 2: Data Objects

A reusable and repurposing LO model should have the characteristics such as cohesion, decoupling, modularity, inheritance and overriding. The modularity of LO-class is achieved with the help of data objects and learning material objects. The learning material objects are raw in nature and simply representing information without any learning objective. These objects may have captions, but they do not associate with any 'meta data'. That is, these raw objects cannot be used directly in 'content aggregation processes'. For example, 'text', 'table', 'picture', 'image' files are some learning materials. A collection of learning materials with a learning objective forms a data object. Data objects are themselves learning objects at a lower level with almost in monolithic form. For example, a general description of 'IntegerArray', a quiz on 'IntegerArray' and a tutorial on 'JavaString' can be data objects.

Data objects are well-formed building elements to construct LO-classes.

Each data objectas shown in Figure 2 has a set of Learning Material elements with a metadata element. Learning Material elements in a data object include one introduction element, one or more subject explanation elements and one evaluation element. The axiom of our approach circulates around the subject whereby to increase the understanding the capability of LO concept. In addition to this, we formulate a layered approach in organizing LO structure which is aimed for repurposing and exploiting the use of LOs.

IV. LO CLASS HIERARCHY

The Object-Oriented data modelcreatessimply a LO class hierarchy. Here, we use the term 'LO class' to mean either a LOtemplate or a LO object. Hence, the hierarchical tree denotes derivation of readily usable LO objects called concrete LO classes from abstract level templates called LO abstract classes. A concrete LO class is treated as LO object. The name'interface'(java's terminology) can be used here to denote a template. An interface simply narrates the content definition, but no content information. The interfaces must be implemented before they can be used or reused as LO objects. The root LO class is an interface class. The LO hierarchy starts with the root interface class. Many interface subclasses can be derived from a parent interface class using inheritance operation. An implemented interface becomes aconcrete LO class that is readily usable or reusable in more than one learning context.In general, inheritance helps to either reuse or repurpose the content types. Several levels of inheritance may be needed to repurpose the contents that are applicable to various contexts, but with a diminished scope of reusability. Deriving an effective LO from anabstract LO-type is called ‘repurposing’. Ultimately the ‘repurposing’is achieved with the help of object-oriented framework.Figure 3 shows a general picture of LO-hierarchy in this framework.

An LO-hierarchy represents a subject domain. Assuming that a subject is described using a set of LOs, the leaves of a class hierarchy represent the fully implemented LO classes. There is no fixed guideline to define a subject domain. The domain is selected as large one in order to find a high degree of reusing and repurposing within that domain. The ‘C language family’ domain is a good example where many LOs can be reused or repurposed among C, C++, Java, C-sharp, etc. Similarly, operating system family, database family, etc are also independent subject domains. These domains can be implemented as subdomains of the computer science domain. The hierarchy thus keeps growing until a super root called global root.

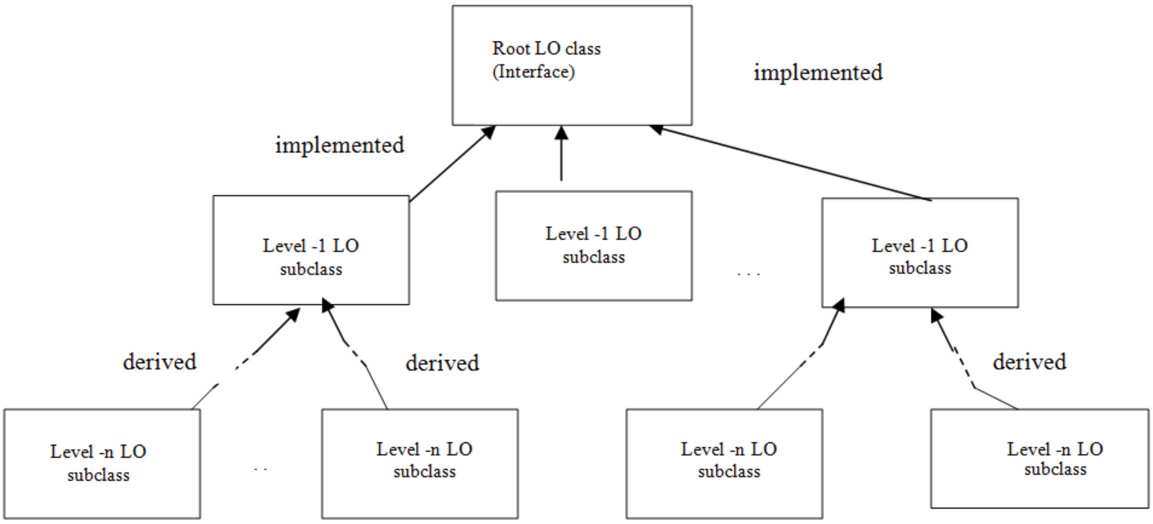

Figure 3: LO Class Hierarchy

V. DESIGNING LO CLASS HIERARCHY

For our experiment the LO-hierarchy for C language family subdomain is built as shown in Figure 4. At first list the concepts that are going to be implemented as LOs. In C language family the list of concepts include ‘program’, ‘data types’, ‘variables’, ‘constants’, ‘functions’, ‘statements’, methods, classes, iostreams, etc. Then the common learning content elements of these concepts become the definition of the root interface. The common learning content elements of C language family include general introduction of the concept, formal definition of the concept, syntax description of the concept, examples, tutorial, etc.The root interface class called CLFamilyatthe root describes the minimum LO content definition for the C languagefamily concept.

The object-oriented framework explained in this example uses both inheritance and implementation to create the C Language family hierarchy. The inheritance of interface at the root would allow augmenting the inherited content definition for each language concept. The implementation of an interface helps to create LO objects with contents. The inheritance mechanism can be used either as reusable contents in another learning context or as modified contents for repurposing. The inheritance here is same as the ‘is-a’ relationship. All derived or inherited LO classes are children (i.e. subclasses of a parent LO class) or grandchildren (i.e. sub-subclasses of a parent LO class).

A good example of reusability and repurposing is shown in Figure 5. In this example an LO-class ‘Array’ is an abstract level or a more generic class that can be used in many learning contexts of programming language domain as an introductory part. Several context-based LO-classes such as ‘CArray’, ‘CPPArray’, ‘JavaArray’, etc can be derived as specific instances. The LO-subclasses naturally inherit all characteristics or properties from their parent LO-class. Figure 5 shows the LO hierarchies for Array-class.

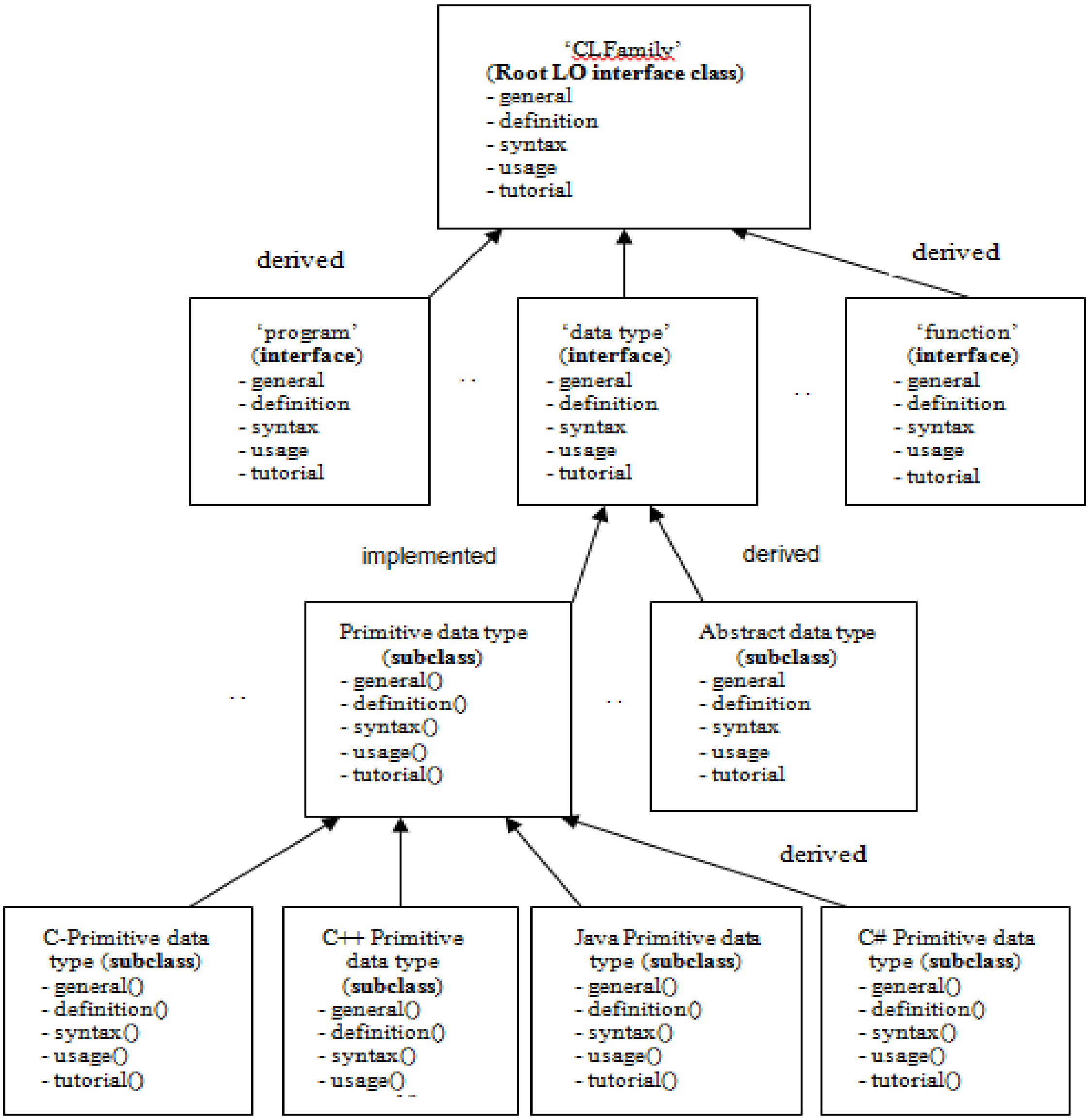

Figure 4: LO Class hierarchy implementation for C Language Family

The inheritance mechanism creates specialized LO-classes (i.e. LO-subclasses) from the existing LO-classes by adding required properties (i.e. data or information elements) suitable for repurposing towards the specified learning contexts. Here, the inherited information is reused in the derived LO-class. The derived LO-classes have new data (or information) elements that are not part of their parent LO-class. For example, CArray LO-class might add C-specific data elements such as "CIntegerArray", "CRealArray", etc. Similarly the JavaArray might add new Java-specific information elements such as Java1DArray, Java2DArray etc. These added information elements are not found in their parent LO-classes.

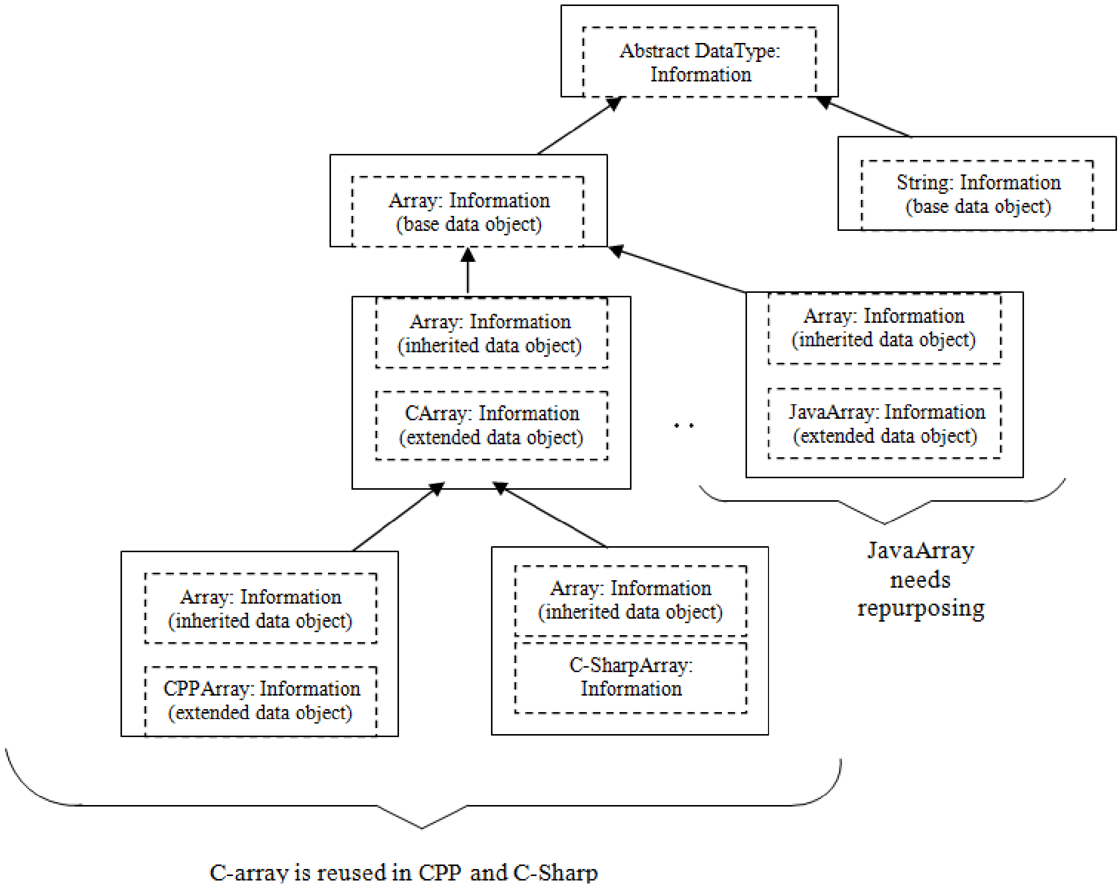

Figure 5: Hierarchy showing both reusable and repurposing

The two important properties of software engineering namely 'cohesion' and 'decoupling' help the LO-class model in catering a precise and a concise learning. Cohesion refers to the degree in which the information elements within an LO-class form a single, unified concept, with no excess elements. A strong cohesion means easier comprehension and thus more reliable and precise knowledge [26]. The class hierarchy indirectly indicates the various levels of cohesive property. The LO-class at the root level has minimum cohesiveness whereas the leaves of the tree have maximum cohesiveness. The decoupling refers to independent nature of LO-classes. The least dependent LO-classes are insensitive with respect to any changes in other LO-classes in the same domain. This helps LO-class maintenance easier.
The information hiding is one of the essential characteristics of Object-Oriented programming [27]. It helps maintaining the data integrity. There is no need for information-hiding with LO classes. Instead the inheritance tree implies a gradual exposure of detailed information. That is, the children expose more detailed information than their parents' information. Down the hierarchy the classes are more and more specialized and the sametime detailed information is more and more exposed. Overriding property can also be found useful in deriving a closely related LO-class from its parent. Overriding helps to redefine the existing information with new information.

VI. VI. CONCLUSION

Our investigation into various LO models identify a need for a structured approach to develop LOs that can be reusable and reconfigurable for repurposing in a smooth manner. In order to achieve this goal we have developed a class-based LO framework closely following Ed Morris work [9] using object-oriented programming discipline. The framework can be used to design a domain-based LO hierarchy for a user designed pedagogical scheme. Another advantage of this model is that the resulting LO repository is of hierarchical nature allowing efficient LO searching.

As a future work, the current researchers would like to use the proposed framework in different e-Learning contexts to check the feasibility of the repurposing property.

REFERENCES

[1] L'Allier, J. (1997). *Frame of Reference: NETg's Map to Its Products, Their Structure and Core Beliefs*. [Electronic version].

Retrieved April 1, 2014, from http://www.im.com.tr/framerefer.htm.

[2] Cisco System, Inc. (1999). *Cisco systems Reusable Information Object Strategy: Definition, Overview and Guidelines*. Retrieved June 4, 2014, from http://www.cisco.com/warp/public/779/ibs/solutions/learning/whitepapers/el_cisco_rio.pdf

[3] Hodgins, H. (2000). *The future of learning objects*. In D. A. Wiley (Ed.), The Instructional Use of Learning Objects: Online Version. Retrieved July 5, 2014, from: http://www.reusability.org/read/chapters/hodgins.doc

[4] Wiley, D. (2000). *Learning Object Design & Sequencing Theory*. Retrieved March 14, 2014, from http://www.opencontent.org/docs/dissertation.pdf

[5] Wiley, D. (2001). *Connecting Learning Objects to instructional design theory: A definition, a metaphor, & a taxonomy, Retrieved* March 7, 2014, http://www.elearning-reviews.org/topics/technology/learning-objects/2001-wiley-learning-objects-instructional-design-theory/

[6] Boyle, T. (2002). *Design Principles for Authoring dynamic, reusable Learning Objects, Winds of Change in the Sea of Learning: Proceedings 19th ASCLITE* Conference, pp. 57-64, Aukland, New Zealand.

[7] IEEE 2002, (2002). *Draft Standard for Learning Technology - Learning Object Metadata*. Institute of Electrical & Electronic Engineering IEEE P1484.12.1-2002. Retrieved August 1, 2014, from http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf.

[8] Polsani, P. (2003). *Use and Abuse of Reusable Learning Objects*. Journal of Digital Information, 3(4), Article No. 164.

[9] Morris, E. (2005). *Object Oriented Learning Object*, Australasian Journal of Educational Technology, 21(1), pp. 40-59.

[10] Jones, R. and Tom Boyle (2007). *Learning Object Patterns for Programming*, Interdisciplinary Journal of Knowledge and Learning Objects, 3, 2007.

[11] Wang, T. and Chang, F. (2007). *Tamkang University, SCoRM Compliant Courseware Authoring Tool for Supporting Pervasive Learning*,International Journal of Distance Education Technologies, 5(3), 2007, IGI Global.

[12] Ramanathan, C. (2011). *A System for Managing SCORM-Compliant Learning Objects*. IEEE International Conference on Technology for Education (T4E), Banglore, India.

[13] Dimitriadis, Y., McAndrew, Conole, G. and Makriyannis, E. (2009). *New design approaches to repurposing open educational resources for collaborative learning using mediating artefacts*, Proceedings of Ascilite 2009Auckland.

[14] Al-Khanjari, Z., Kutti, N. and Al-Mahri, F. (2010). *RMLOM: Reusable Multipurpose Learning Object Model, Proceedings of The 4th International Conference on Information Systems*, Technology and Management (ICISTM-10), March 11-13, 2010, Bangkok, Thailand.

[15] Idrosa, S., Mohameda, A., Esaa, N., Samsudina, M and Dauda, K. (2010). *Enhancing Self-directed Learning Skills through e-SOLMS for Malaysian Learners*, Procedia Social and Behavioral Sciences, 2(2): 698-706.

[16] Pegler, C. (2011). *Reuse and Repurposing of Online Digital Learning Resources within UK Higher Education: 2003-2010. PhD thesis*, The Open University, UK.

[17] Sampson, D. and Zervas, P. (2011). *A Workflow for Learning Objects Lifecycle and Reuse: Towards Evaluating Cost Effective Reuse*. Educational Technology and Society, 14(4): 64-76.

[18] Kaldoudi, E., Dovrolis, N., Konstantinidis, S. and Bamidis, P. (2011). *Depicting Educational Content Repurposing Context and Inheritance*, IEEE Transactions on Information Technology in Biomedicine, 15(1), pp. 165-170, January 2011.

[19] Holz-Clause, M. and Guntuku, D. (2012). Journal ofSystematics, Cybernetics and Informatics, 10(4): 100-103.

[20] Rufer, R. and Adams, R. (2013). *Deep learning through reusable learning objects in an MBA program*. Journal of Educational Technology Development and Exchange, 6 (2), 69-82.

[21] Minovic, M., Milovanovic, M. and Starcevic, D. (2013). *Learning Object Repurposing for Various Multimedia Platforms, Multimed Tools Appl, 63:* 927-946.

[22] Sultan, T., Nasr, M. and Amin, S. (2014). *Learning Objects Reusability Effectiveness Metric (LOREM), Research Journal of Applied Sciences*, Engineering and Technology, 7(12): 2475-2482.

[23] Piedra, N., Chicaiza, J. and Lopez, J. (2014). *An Architecture Based on Linked Data Technologies for the Integration and reuse of OER in MOOCs Context, Open* Praxis, 6(2): 171-187.

[24] Hadjerrouit, S. (2007). *Applying a System Development Approach to Translate Educational Requirements into E-Learning, Interdisciplinary Journal of Knowledge and Learning Objects*, Vol. 3, 2007.

[25] Chimalakonda, S. and Nori, K. (2012).*A Software Engineering Perspective for Accelerating Educational Technologies*, 2012 IEEE 12th International Conference on Advanced Learning Technologies (ICALT), January 2012.

[26] Sommerville, I. (2011). *Software Engineering, 9th ed., Addison-Wesley.*

[27] *Nicholas*, C. (2014). Principles of Object-Oriented Programming in JavaScript, William Pollock, USA.