# E-learning Materials Development: Applying Software Reuse Principles and Granularity Levels in the Small

Nabil Arman*
Department of Information Technology
Palestine Ahliya University
Bethlehem, Palestine
Email: narman@paluniv.edu.ps

***Abstract*— The development of e-learning materials is typically acknowledged as an expensive, complicated, and lengthy process, often producing materials that are of low quality and difficult to adapt and maintain. It has always been a challenge to identify proper e-learning materials that can be reused at a reasonable cost and effort. In this paper, software engineering reuse principles are applied to e-learning materials development process. The reuse of existing e-learning materials is beneficial in improving developers of e-learning materials productivity. E-learning material reuse is performed, in this research, based on construct's granularity rather than on unified constructs of one size.**

## I. INTRODUCTION

Developing appropriate e-learning materials is an expensive, complicated, and lengthy process. Therefore, the reuse of existing e-learning material is of great value. To reuse existing e-learning materials, one has to locate and sometimes adapt such materials. Many approaches have been suggested in the literature for reuse of existing e-learning materials, but these approaches are tailored for reuse in the large [1,2]. Some approaches advocate Advanced Distributed Learning (ADL) Sharable Content Object Reference Model (SCORM) - A suite of standards and specifications for online education that enables interoperability of learning content. SCORM implements a modular approach to online learning that aggregates discrete units of digital instruction called learning objects. Learning objects are self-contained and may be reused in multiple contexts and environments, including online courses, knowledge management systems, and performance support systems [3]. Despite SCORM's evident advantages, taking that into account proved to be expensive and beyond the capabilities of e-learning materials reuse initiatives in the small.

A three-layer e-learning course development model has been suggested in [2]. The model starts by decomposing the learning content into small chunks placed in a hierarchic structure of units, blocks, sub-blocks, …etc. Based on these small chunks learning objects (LO) are obtained. These LO are candidates for reuse.

In this paper, a new approach is presented for reuse of e-learning materials in the small. The new approach splits the e-learning material into smaller units that can be placed in a hierarchy consisting of large units at the top of the hierarchy and smaller units are placed at lower levels of the hierarchy allowing reuse of units at different granularity levels. These units can be re-used easily in a more productive manner.

## II. SOFTWARE REUSE: BENEFITS, PROBLEMS, AND PRINCIPLES

Although software reuse has many evident benefits, it has also many problems. These benefits and problems can be summarized as follows [4,5]:

a. Increased dependability
b. Reduced process risk
c. Effective use of specialists
d. Standards compliance
e. Accelerated development
f. Improvements in productivity
g. Improvements in quality

The problems of software reuse can be summarized as follows [4]:

a. Increased maintenance costs
b. Lack of tool support
c. Not-invented-here syndrome
d. Creating and maintaining a component library
e. Finding, understanding and adapting reusable components

The benefits and problems of software reuse do apply for the reuse of e-learning materials. However, the effects of the problems may be minimized by concentrating on the most important aspects and attributes of e-learning units. The use of sound reuse principles can also minimize the effects of problems and reduce greatly the associated costs as explained later.

---

* Dr. Arman is on a sabbatical leave from Palestine Polytechnic University during 2009/2010 academic year

As for the software reuse principles, a set of candidate principles for software reuse is suggested in [6]:

1. Build a software domain architecture as a framework for reuse activities.
2. Use a software development process that promotes and controls reuse.
3. Reuse more than just code.
4. Practice domain engineering.
5. Integrate reuse into project management, quality management and software engineering activity.
6. Organize the enterprise to facilitate partnering to achieve reuse across product boundaries.
7. Use automation to support reuse.
8. Couple modern reuse theory and technology with natural, traditional organizational reuse practices.

## III. REUSABLE E-LEARNING CONTENT/MATERIALS

Before describing the reuse of e-learning materials for a new course, one has to determine the parts of the materials of the course that should be taken into account. These may include:

1. Syllabus: Although, parts of the syllabus can be reused like intended learning outcomes (ILOs) and the list of contents of a course, it does not represent a major construct in our suggested approach. There is not that much saving in reusing the syllabus and the time needed to prepare a syllabus is very short when compared to the time and effort needed to prepare the e-learning material itself.

2. Contents: The course contents consist of a number of chapters that are divided into modules. Modules, in turn, are divided into units. Units may also be divided into subunits, …etc. These small subunits may include text segments, exercises, questions, data, figures, keywords, project descriptions, and so on. Considering different levels of granularity, a complete chapter may be very large to be reused as a whole and a subunit may be considered too small. In addition, these components may be obtained from different sources. The components may be arranged in a whole-part UML aggregation that depicts special relationships between the components as shown in Figure 1.

As shown in Figure 1, an e-learning material developer should start looking at complete chapters as a reusable construct. If that is not possible, one should consider modules, then units, then subunits, …etc.

Other approaches, however, use unified reusable constructs of the same size. Thus, for a complete chapter to be reused, many smaller reusable constructs are assembled together to form the chapter being reused.

This is a costly and error-prone process since it involves many iterations of the reuse process testing the appropriateness of the reusable construct under consideration to be reused in the new chapter.

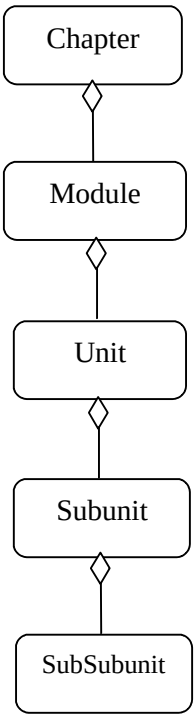

Figure 1. UML aggregation of e-learning constructs

## IV. REUSABLE E-LEARNING MATERIAL DEVELOPMENT APPROACH

A reusable construct should have a number of attributes that are essential in determining the appropriateness of this construct in the reuse process. These attributes, which are considered as metadata for the e-learning constructs, include:

ID: The identification number of the construct
Name: The name of the construct
Short Description: A short description of the construct
Learning Outcome: A brief description of the learning outcome
Language Used: The language used in the construct
Format: The format of the construct
Duration/Length: The time needed to complete the construct
Level: The intended level of the construct
URL: Uniform resource locator of the construct
Whole ID: The ID of the parent of the construct

Certain constructs may be reused from different sources. Therefore, these constructs may be arranged in a way to avoid any conflicts that may result from such a situation.
Similar to software reuse, e-learning materials reuse consists of processes such as: identifying reusable constructs, describing the constructs, retrieving reusable constructs, adapting retrieved constructs to specific needs, and integrating constructs into the e-learning material being developed [6]. In software reuse, these processes are complicated, made more complicated by large number of reusable components with which a software engineer must deal with. This doesn't apply in our case since we are dealing with e-learning material reuse in the small. This means that we are dealing with e-learning

material reuse in a small organization like a university or a consortium of universities and academic institutions.

Based on the circumstances and the degree of similarity, based on the constructs between the e-learning material being developed and the existing e-learning material, a complete chapter may be reused as a whole and in other circumstances a subunit may be reused, based on the meta data associated with the e-learning constructs.

In developing a new e-learning course and after taking the syllabus of the course into account, one should start by considering the possibility of reusing a whole chapter. If that is not possible, consider reusing a module, then consider reusing a unit, and so and so forth. The larger the reused component the better the situation is.

In our approach, the principles of software reuse are taken into account. In the e-learning material development process, reuse is promoted by associating meta-data for all components (chapters, modules, units, subunits, …etc.). Another principle is domain engineering, which is defined as the entire process of reusing domain knowledge in the production of new software systems, is practiced in the development of reusable e-learning material. This is very beneficial since domain engineering is a key concept in systematic software reuse. A key idea in systematic software reuse is the application domain, a software area that contains systems sharing commonalities [8]. Taking domain engineering into account, one should consider reusing e-learning materials from the same domain, in terms of the subjects and the organizations.

As stated in the principles of software reuse, automation that supports e-learning material development is very beneficial especially in the process of listing the set of candidate e-learning constructs to be reused.

In addition, integrating reuse into quality management activity is very important, since reusable e-learning material is likely to be of high quality.

Using the meta-data for the e-learning constructs, one can manually identify the suitable e-learning constructs for reuse. As mentioned before, one should always strive to reuse larger constructs. Instead of performing adaptations for the larger constructs, one can proceed to smaller constructs.

The next step in this research effort is the automation of the process of e-learning material reuse. Instead of searching for candidate reusable constructs manually, one may consider developing an algorithm for identifying the most appropriate reusable e-learning construct based on its meta data. In addition, one may develop an approach for obtaining the input for such a tool from the e-learning platform used. Getting such input from an open-source tool like MOODLE is easier since its backend database schema is easily accessible.

Regarding the automation of the process of e-learning material reuse, a taxonomy of subjects and their sub-subjects is currently being investigated to determine the list of e-learning constructs that are candidates for reuse. Of course, one can not list all subjects and sub-subjects, therefore, search by keywords is another possibility.

## V. CASE STUDY

A database Systems course can be found in many information technology related programs like computer science, computer systems engineering, information technology and information systems. A typical set of topics to be covered in a computer science database systems course includes:

| | |
|---|---|
| Chapter 1: Database Concepts and Architecture | 1.1 Introduction<br>1.2 An Example<br>1.3 Characteristics of the Database Approach<br>1.4 Advantages of the Database Approach<br>1.5 Data Models, Schemas, and Instances<br>1.6 DBMS Architecture and data independence<br>1.7 Database Languages |
| Chapter 2: Data Modeling Using the Entity-Relationship Model | 2.1 Conceptual Data Models<br>2.2 An Example Database Application<br>2.3 Entity Types, Attributes, Keys<br>2.4 Relationship Types and Constraints<br>2.5 Weak Entity Types<br>2.6 Data Modeling CASE/Diagramming Tools |
| Chapter 3: Normalization for Relational Databases | 3.1 Informal Design Guidelines for Relation Schemas<br>3.2 Functional Dependencies<br>3.3 Normal Forms Definition<br>3.3.1 First Normal Form<br>3.3.2 Second Normal Form<br>3.3.3 Third Normal Form<br>3.3.4 Boyce-Code Normal Form<br>3.3.5 Fourth Normal Form<br>3.3.6 Fifth Normal Form<br>3.3.7 Domain Key Normal Form |

A typical set of topics to be covered in an information systems database systems course includes:

| | |
|---|---|
| Chapter 1: Database Concepts and Architecture | 1.1 Introduction<br>1.2 An Example<br>1.3 Characteristics of the Database Approach<br>1.4 Advantages of the Database Approach<br>1.5 Data Models, Schemas, and |

| | |
|---|---|
| | Instances<br>1.6 DBMS Architecture and data independence<br>1.7 Database Languages |
| Chapter 2: Data Modeling Using the Entity-Relationship Model | 2.1 Conceptual Data Models<br>2.2 An Example Database Application<br>2.3 Entity Types, Attributes, Keys<br>2.4 Relationship Types and Constraints<br>2.5 Weak Entity Types |
| Chapter 3: Normalization for Relational Databases | 3.1 Informal Design Guidelines for Relation Schemas<br>3.2 Functional Dependencies<br>3.3 Normal Forms Definition<br>3.3.1 First Normal Form<br>3.3.2 Second Normal Form<br>3.3.3 Third Normal Form<br>3.3.4 Boyce-Code Normal Form |

If the computer science database systems course was prepared as a reusable e-learning material with meta-data associated to chapters, modules, units, and subunits as suggested in our approach, then preparing the e-learning material of the information systems database systems course can be accomplished as follows:

1. Chapter 1 from the computer science database systems course can be reused as a whole. The granularity of reuse here is a complete chapter.

2. Modules 2.1, 2.2, 2.3,2.4, and 2.5 from the computer science database systems course can be reused. However, module 2.6 is not reused. The granularity of reuse here is a module.

3. Modules 3.1 and 3.2 are reused. From module 3, units 3.3.1, 3.3.2, 3.3.3, and 3.3.4 are reused. The granularity of reuse here is a module and a unit.

Thus, instead of decomposing the e-learning material into learning object of small granularity, it is suggested in this research to decompose the e-learning material into learning constructs of different granularity levels and arrange these constructs in a UML-like aggregation to facilitate the reuse process and to improve e-learning materials developers' productivity. As mentioned in the principles of software reuse, this should be promoted in the e-learning development process itself. Of course, the larger the reusable constructs used, the lower the cost of the e-learning material development.

## VI. CONCLUSIONS

A new approach that utilizes principles of software reuse is applied to the development of e-learning materials is presented. The approach suggests reusing e-learning constructs of different granularity levels to improve e-learning material developers' productivity. A case study that demonstrates the benefits of this approach is presented.

## REFERENCES


[1] Jancso, T., and Markus, B. "Reuse and Sharing of E-learning Materials Inside the EU," FIG Working Week 2007, Hong Kong SAR, China, 13-17 May, 2007.

[2] Schreurs, J. and Dalle, R. "Authoring Systems Delivering Reusable Learning Objects," iJET, Vol. 4, Special Issue 2, October 2009.

[3] Hersh, W. and Bhupatirajum R, "Adopting e-Learning Standards in Health Care: Competency-based Learning in the Medical Informatics Domain", AMIA Annual Symposium Proceedings, pp. 334–338, 2006.

[4] Sommerville, I. Software Engineering, 8th Edition, Addison-Wesley, 2008.

[5] Grabert, M. and Bridge, D. "Case-Based Reuse of Software Examplets," Journal of Universal Compter Science, Vol. 9, No. 7, pp. 627-640, 2003.

[6] James, M. "Fundamental Principles of Software Reuse," Eighth Annual Workshop on Institutionalizing Software Reuse (WISR), Ohio State University, Ohio, USA, March 23-26, 1997.

[7] Smolarova, M. and Navrat, P. "Software Reuse: Principles, Patterns, Prospects," Journal of Computing and Information Technology, Vol. 5, No. 1, pp. 33-49, 1997.

[8] Frakes, W. and Kang, K. , "Software Reuse Research: Status and Future", IEEE Transactions on Software Engineering, Vo. 31, No. 7, pp. 529-536, 2005.