Chapter 6  Framework for Learning Content Management

6.1  Existing Scenarios

In many parts of the world, dissatisfying experience in information and knowledge services available at the Rural Knowledge Centres (RKC) [136], have induced search for more appropriate ways to deal with the complexity of the issue. This is because, till the beginning of this decade more focus was given on providing the technological infrastructure to help people get connected, but not so much on relevant content [137]. With increasing unsatisfactory experiences, in recent years, in the implementation of information and knowledge services put the development of appropriate and locally relevant content as a major issue in the RKC projects [138], [139]. Several other studies have also pointed out the importance of locally relevant content in spreading the benefits of Information and Communication Technologies (ICTs) widely among local communities [140], [141]. As a result different technology mediated approaches were emerged for satisfying the learner needs such as delivery of content via internet, intranet/extranet, audio and videotape, satellite and CD-ROM. Such applications are called as open and distance learning approaches i.e., computer-based learning, web-based learning, virtual classrooms and digital collaboration (discussed in detail in section 1.5.1).
The contemporary ICTs, specifically Multimedia, Networking, and Software Engineering have promoted the apparition of a huge amount of learning resources. However, most of these approaches are context and technology specific, which many times demand content designers and developer to rework on the existing content again, to take little modifications, considering the users need and technical preferences. This has been viewed as an expensive and time consuming process. Hence development of content and enables it to be packaged as Reusable Learning Objects (RLOs) is carving a new path for research on reusing, and repurposing content.

It was therefore, the research work includes to develop a framework for rapid generation of locally relevant and appropriate content from a generic available Reusable Learning Objects (RLOs) pool, and its management in an affordable way at minimal required infrastructure.

### 6.1.1 Reusable Learning Objects

Reusable Learning Object (RLO) is a specific chunk of digitized instructional content that can be reused in many different learning content modules. According to Dahl and Nygaard [142], Learning Objects are elements of a new type computer-based instruction grounded in the object-oriented paradigm of computer science; Object-orientation highly values the creation of components (called “objects”) that can be reused in multiple contexts. Reigeluth and Nelson [143] suggest that when teachers first gain access to instructional materials, they often break the materials down into their constituent parts, and then reassemble these parts in ways that support their individual instructional goals. This is the fundamental idea behind RLOs: instructional designers can build instructional components that can be reused number of times in different
learning contexts. These RLOs can be tailored according to the specific needs and deliverable over the Internet, meaning that any number of people can access and use them simultaneously. Moreover, those who incorporate learning objects can collaborate on and benefit immediately from new versions.

To facilitate the widespread adoption of the learning objects approach, the Learning Technology Standards Committee (LTSC) of the Institute of Electrical and Electronics Engineers (IEEE) formed in 1996 to develop and promote instructional technology standards [144]. Without such standards, universities, corporations, and other organizations around the world would have no way of assuring the interoperability of their instructional technologies, specifically their RLOs.

6.2 Existing Technologies and Standards

The learning technology standardization process is an active, continuously evolving process that will last for years to come, until a clear, precise, and generally accepted set of standards for educational-related systems is developed. The main contributors to this effort are the IEEE's Learning Technology Standardization Committee (LTSC) [145], the IMS Global Learning Consortium [146], the Aviation Industry CBT Committee (AICC) [147], the US Department of Defense's Advanced Distributed Learning (ADL) initiative [148], and projects Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE) [149], Getting Educational Systems Talking Across Leading Edge Technologies (GESTALT) [150], PROmoting Multimedia access to Education and Training in EUropean Society (PROMETEUS) [151], European Committee for Standardization, Information Society Standardization System,
Learning Technologies Workshop (CEN/ISSS/LT) [152], Gateway to Educational Materials (GEM) [153], and Education Network Australia (EdNA) [154]. The IEEE's LTSC is the institution that is actually gathering recommendations and proposals from other learning standardization institutions and projects. Specifications that have been approved by the IEEE go through a more rigorous process to become ANSI or ISO standards.

Although many standards are available Sharable Content Object Reference Model (SCORM) specifications has been widely used. In this study SCORM specifications has been examined and adapted for developing a framework to reuse (or) repurpose learning content and its management.

### 6.3 Sharable Content Object Reference Model (SCORM)

Sharable Content Object Reference Model (SCORM) is a suite of technical standards and speciation given by ADL, that enable web-based learning systems to find, import, share, reuse, and export learning content in a standardized way. The purpose of SCORM is to achieve interoperability, reusability, accessibility and durability.

The SCORM is a conceptual model describing how to manage, package and deliver learning information so that it can be easily shared on the Internet. This means that if a course is packaged adhering to the SCORM, this course can be shared with other people. In addition, parts or sections of the course can be shared with other people without having dependencies on other parts of the course. According to the SCORM specifications, this is assuming that there is a SCORM implementation available and that the course to be shared has been packaged correctly.
SCORM is an initiative funded by the US military and was started with the desire to train military personnel all over the world on different platforms easily. The Department of Defence (DoD) established the Advanced Distributed Learning (ADL) initiative in 1997 to develop a DoD wide strategy for using learning and information technology to modernize education and training. They released the first version of SCORM 1.0 on 31st January 2000 named as Sharable Courseware Object Reference Model. They updated this version and released SCORM version 1.1 on 16th January 2001 and changed the name from courseware to Sharable Content Object Reference Model and they released SCORM version 1.2 on 1st October 2001 by introducing the content packaging idea. They released SCORM version 1.3 on 31st January 2004.

ADL initiative has defined high-level requirements for learning content to leverage existing practices, promote the use of technology-based learning and provide a sound economic basis for investment.

Accessibility
The ability to locate learning content in multiple locations and deliver it to multiple locations.

Interoperability
Operate across a wide variety of hardware, operating systems and web browsers.

Durability
The ability to withstand technological changes (or) do not require significant modifications with new versions of system software.

Reusability
The flexibility to deploy content in multiple applications.
The main advantage of SCORM is that it allows for sharable learning content. For example a lecturer was running a course on OpenACS and in that course was a lesson on TCL, someone else could include this section in their course on, say, scripting languages. This can be accomplished due to the strict guidelines for packaging a course, as well as the standard Run-time Environment Model across all platforms.

SCORM includes both a **Content Aggregation Model** and a **Run-time Environment Model**. The Content Aggregation Model has several parts, the content model, metadata, content packaging and a sequencing model. These parts describe the necessary contents of the learning information, how to describe it using metadata, how to package it all together and how to sequence it. The Run-time Environment Model explains how to start learning resources, the mechanism for learning resources to communicate with an LMS (Learning Management System) and the language forming the basis for communication.

### 6.3.1 **SCORM Content Aggregation Model**

The content aggregation model explains the process of creating, discovering and gathering together simple assets into complex sharable learning objects and organising those learning objects into a predefined sequence for delivery. This section gives a brief explanation of these processes, the official SCORM documentation goes into more detail.
**SCORM Content Model Components**

The Content Model Components are the base level parts of a course. The SCORM defines three types of content model components, Assets, SCOs and SCAs. Each of these components must be completely self-sufficient in order to allow sharing ability without any other dependencies.

**Asset**

An Asset is the simplest form of learning content. An Asset is any file that can be delivered to a client browser, for example an image, video file, audio file or an XML document.

**Sharable Content Object (SCO)**

An SCO is a collection of one or more Assets that use the SCORM Run-Time Environment to communicate with a Learning Management System (LMS).

**Sharable Content Asset (SCA)**

The SCA is a new addition to the SCORM, it was added as a new learning object in SCORM version 1.3. Like an SCO, an SCA is a collection of Assets forming a sharable learning resource which can be launched by an LMS. The one simple difference between an SCO and an SCA is that an SCA makes no calls to a LMS via the API Adapter. This means that there is no need for the SCA to be dynamic or store any data. An SCA could be something like a lesson composed of a standard HTML page containing images.
Content Aggregation

Content Aggregation is a map which links activities together. An Activity is a collection of, or individual SCOs or SCAs (it could be a course, module, chapter, lesson etc.). An Activity can contain other Activities, which contain other Activities and so on. For example a course could contain chapters, which contain sections, which contain quizzes and lessons. A base level Activity is either an SCO or an SCA.

SCORM Meta-data Components

This section describes meta-data and explains how it is used to describe SCORM Content Model Components to allow for component searching and reuse. The official SCORM specifications give further detail on applying meta-data to learning resources. Meta-data is data about data; it provides a common way for learning resources to be described. Learning resources that have been described by meta-data can be searched for and retrieved for use and reuse. The SCORM provides guidance for applying meta-data to Assets, SCOs, SCAs, Activities and Content Aggregations. Following is a description of the different types of meta-data for each of the Content Model Components.

Asset Meta-data provides descriptive information for an Asset. This allows for search and discovery of that Asset from within repositories.
**SCO and SCA Meta-data** describes an SCO or SCA (being a collection of Assets) in order to allow for discoverability of that SCO or SCA from within a repository. These two types of meta-data are essentially the same due to the similarity between SCOs and SCAs.

**Activity Meta-data** is a new addition to the SCORM as of version 1.3. Activity Meta-data describes the Activity as a whole and is intended for discoverability of that Activity from within a repository.

**Content-Aggregation Meta-data** describes the structure of the Content-Aggregation as a whole. This meta-data is intended for the discoverability of Content-Aggregations from within a repository.

The SCORM Meta-data Information Model is broken up into nine categories.

1. The **General** category groups the general information that describes the resource as a whole.
2. The **Lifecycle** category groups the features related to the history and current state of this resource and those who have affected this resource during its evolution.
3. The **Meta-metadata** category groups information about the meta-data record itself (rather than the resource that the record describes).
4. The **Technical** category groups the technical requirements and characteristics of the resource.
5. The **Educational** category groups the educational and pedagogic characteristics of the resource.
6. The **Right** category groups the intellectual property rights and conditions of use for the resource.

7. The **Relation** category groups features that define the relationship between this resource and other targeted resources.

8. The **Annotation** category provides comments on the educational use of the resource and information on when and by whom the comments were created.

9. The **Classification** category describes where this resource falls within a particular classification system.

**Content Packaging**

Content Packaging defines how learning content should be packaged on disk to facilitate sharing learning resources. Basically a packaged course is stored in the form of a `.zip` file in a specific way. Each `.zip` file has to include an `IMSManifest.xml` file that explains how the whole course fits together. There are many SCOs (Sharable Content Objects) in a course; they are basically sections of a course, or lessons. Each of these SCOs needs to be completely independent, so that it can be used in other courses without having any other dependencies.

The IMSManifest file is an XML file that maps each Asset with each SCO or SCA and defines the order in which the SCOs and SCAs are to be delivered. The SCORM defines the structure of the XML file and details how the course should be packaged in order to deliver a complete learning experience.
The role of the Run-time Environment is to deliver learning content to the user in the correct order and allow the learning content to store and retrieve data. The JavaScript API is the mechanism for which the Run-time Environment (RTE) does this.

Through the API, SCOs make calls to the LMS. The LMS responds to the SCO with an appropriate response. The API adapter in this case is a Java Applet embedded in the clients’ browser that facilitates this connection between the SCOs and the LMS.

**Sequencing**

Sequencing is the navigation between learning resources (SCOs, SCAs and Assets). The sequencing is necessary to create a desired course package by tailoring learning resources according to the predefined order. The official SCORM documentation gives a complete description affecting sequencing behaviour in a complete learning experience.

The SCORM Run-time Environment must be able to navigate between learning resources in the correct order. The content aggregation is described by the `IMSManifest.xml` file. Sequencing is more complicated than simply supplying a link from each SCO to the next SCO. The sequencing can be dynamic. If for example the user fails a quiz, they may have to go back and repeat some previous lessons and then take the quiz again. However, if they were to pass the quiz, they would have been passed to the next lesson. Criteria such as this can be stored in the `IMSManifest.xml` file and this will be converted to a content aggregation.

The SCORM Run-time Environment must provide the user with a navigation window that allows for navigation between learning resources. SCOs can also set navigation events so that the user is automatically forwarded to a specified resource after completing that SCO.
It is the Learning Management System’s responsibility to perform the sequencing between learning resources. The LMS must record the location of each user within an Activity and also the Content Aggregation so that it knows what resource comes next. This is implemented by something called a sequencing engine.

### 6.3.2 SCORM™ Run-time Environment Model

This research work did not focus on developing the SCORM Run time environment model directly from the scratch, however the research work develops run time environment by configuring open source LMS and LCMS and test the reusability of learning resources feature with the existing commercial software suits such as Acado. However, in this section the SCORM Run-time environment is examined and discussed briefly to learn about the working mechanism. The official SCORM document gives further detail on applying meta-data to learning resources.

A SCORM compliant LMS is required to implement an API consisting of 8 Functions (Section 3.3 of the SCORM official document provides full details) that content may access to communicate with the LMS.

- `LMSInitialize()`
- `LMSFinish()`
- `LMSGetValue()`
- `LMSSetValue()`
- `LMSCommit()`
This API is implemented by what the SCORM calls an API Adapter. The API Adapter must reside in a window that is a parent window or a parent frame of the window that contains the content. This means that the LMS may launch the content either in a new window or in a frameset. The API Adapter must be an ECMAScript (JavaScript) object named “API” that is accessible through the DOM (Document Object Model). The Adapter must implement the 8 functions listed above.

All communication between the content and the LMS is handled by this adapter, thus the content author does not need to worry about communicating with the server, he/she only needs to be able to find the API Adapter and make the appropriate JavaScript calls. This separation of client and server is essential to the SCORM in that it ensures the portability of content by forcing it to run on a standard platform (the web browser).

For minimal SCORM compliance, the only thing that a piece of content needs to do is call LMSInitialize() when it starts and then call LMSFinish() when it exits. It can be that simple.

In the real-world though, we want a much richer interaction, we want to be able to report test results, track time, bookmark our last location etc. This is where the next 3 functions come in to play. The SCORM defines a data model consisting of data elements which the content can read and write to facilitate this kind of functionality (Section 3.4 of the SCORM official document provides the list of data elements). LMSGetValue() retrieves a data element from the LMS,
LMSSetValue() writes a data element to the LMS and LMSCommit() is called after any values are set to ensure that the data is saved.

The other three functions allow the content to trap and intelligently deal with errors. Implementing this API in the LMS is a little more involved, it has to implement all of the API functions and support most of the SCORM data model. The tricky issue involved with implementing a SCORM compliant LMS is how to handle the browser-to-server communication. Most people choose to do this with a Java applet, but others have been successful using Flash, ActiveX controls and pure JavaScript.

**LMS/LCMS for hosting and delivering the learning objects**

The LMS deals with managing content (sequencing, content packaging) and users (adding users, keeping track of user variables), as well as managing all communication between the content and the users.
6.4 Proposed Framework

Figure 14: Framework for learning content management with RLOs

**Repository:** The repository, in this context, refers to a server/computer where RLOs are organized and stored. The newly created RLOs will be continuously added to this repository in an organized way. In this work, MediaWiki software (an open source software) was configured to develop wiki based content management system.

The Wiki based Content Management System (CMS) was developed at the ICRISAT-KMS department. The XAMPP (Apache Web Server with MYSQL, and PHP) open source software was downloaded from [155], and official installation procedure [156] was followed to configure
the server. The MediaWiki software was downloaded from [157], and official installation procedure [158] was followed to configure the Wiki based CMS (Fig 15).

![Figure 15: Customized Wiki based Content Management System as a Repository for RLOs](http://vasatwiki.icrisat.org/index.php/Drought)

**Content Modelling:** The customized Wiki based CMS allow users to contribute content in any technical formats. However the content needs to be validated and converted into SCORM based RLOs for repurposing, reusing and sharing across multiple locations. The entire process is known as content modeling. The SCORM learning standard specifications (discussed in section 6.3) were followed for content modeling.
Creation of RLOs  An RLO could be created with the available instructional objects in the repository or by new user defined objects. In this work, the pedagogy and SCORM specifications were followed to design and development of RLOs. For instance, *Aphid* is an RLO created with simple assets (two images and text). This Aphid RLO delivers specific instruction on Aphids such as Aphid description, damage symptoms and management of aphids with images (Fig 16).

![Aphid RLO](image)

**Figure 16: Aphid RLO**

Linking RLOs: According to the course organization RLOs could be tailored or linked by writing a code or using any editor such as *RELOAD Editor* (Reusable Elearning Object Authoring and Delivery Editor) [159] or *eXe* (elearning XHTML editor) [160], to name a few.
Method 1:  Customization of RLOs directly from Wiki repository (or) multiple sources

The eXe editor could be used to absorb RLOs directly from Wiki repository. The authors could search and select desired RLOs from Wiki repository and tailor them according to the need to generate a desired learning content package with needed changes. Moreover the eXe allows content from external sources as well, if required. As show in the figure the left pan of eXe has several options to accept content from Wiki and other external sources to create a desired learning content package (Fig 17).

Figure 17: Customization of RLOs directly with eXe editor
Method 2: **Customization of RLOs from an RLOs pool collected from Wiki repository (or) multiple sources with RELOAD editor**

The RELOAD editor provides flexible platform to generate desired learning content package from an RLO pool. As shown in the figure 18, the Reload Editor has three frames (1) resource pan (left frame) (2) manifest pan (right top frame) (3) attribute pan (right bottom frame). The editor allows to open all the available resources from a repository collected from wiki and external resources. The organization could be added for a new course by right click on the organization available at manifest pan, and change the organization name according to the course title by renaming at attribute pan. The RLOs could be drag and drop from resource pan to manifest pan to create a desired learning content package (Fig. 18).

Figure 18: Customization of RLOs with Reload Editor
Learning Consumption: The SCORM compliance learning content packages could be delivered to learners by uploading them into SCORM compliance Learning Content Management Systems (LCMS), Learning Management Systems, Course Management Systems (CMS) (or) develop an API to provide RTE. The complete content package or part of the package can be playable in all the SCORM compliance LMS (or) LCMS (or) CMS.

Re-authoring: The RLOs designed with SCORM specifications lasts for several years. The RLO available in the repository could also be changed according to the need and requirements, which generates a new RLO (Fig. 19). The generation of new RLOs from the available RLOs and content authors designed new RLOs uploaded into the repository allow repository to grow continuously.

Figure 19: Creation of a new RLO from the available RLOs of repository
6.5 Evaluation

The proposed framework was tested, and usability of the framework in the real world scenarios was evaluated. Details are discussed in the following paragraphs.

The SCORM compliance learning content packages were created both on eXe and RELOAD Editor from the resources available on the Wiki RLO repository. These packages were tested on ATutor LCMS (configured an open source software, official installation document was available at [161]) (Fig. 20) and TeN ACADO CMS (configured a commercial software [162]) (Fig 21).

Figure 20: ATutor, A Learning Content Management System
The compatibility of content packages was tested on both ATutor and Acado. The SCORM compliance RLO content packages were downloaded from ATutor and played in ACADO as a new learning content module with minor modifications. For instance, the chickpea learning content package (consists production practices, insect pests, and diseases) was downloaded from ATutor and uploaded in to ACADO for creating a chickpea insect pest course. In ACADO the production practices and diseases sections were deleted, and edited the insect pest sections to create a new course “Chickpea Insect pests”. This approach shows that the SCORM compliance content packages could be playable in any standard open source and commercial CMS (or) LMS (or) LCMS. Since the entire process took only few minutes time, the proposed framework is useful for creating relevant content rapidly by customizing the available RLOs in the repository. The ICRISAT Virtual Academy for the Semi-Arid Tropics adapted this approach for creating their electronic distance learning modules [163].
The user friendliness of the proposed framework was tested in two workshops with scientists of NIH and ICRISAT (Appendix I - 9 participants); and with State Agricultural University faculty (Appendix II - 23 participants). The focus of the workshops was to give them an orientation on content packaging and development of CMS, LMS and LCMS; and hands on session to execute the proposed framework. In both the workshops, the participants were requested to take the exercises individually and in small groups. At the end of the workshop feedback was collected on their comfort levels in executing the framework. The standard instrument developed by the Knowledge Management Sharing department of ICRISAT was used to collect the feedback from the workshop participants.

6.5.1 Observations

In both the workshops, the participants expressed their ease to develop content packages with RELOAD and eXe editors, and uploading them into LMS. One scientist in workshop 1 and three faculty members of agricultural universities in workshop 2 expressed that the designing of RLOs require command on domain knowledge. A professor from Tamil Nadu Agricultural University of Coimbatore said that the techniques taught in the workshop are useful and have shown a direction to resolve content management issues what I had been facing for long time in executing open distance learning courses. Another participant stated that the proposed framework is very useful for sharing knowledge resources among agricultural universities irrespective of technology and systems what they are using.