Formalization of Higher-level Intelligence through Integration of Intelligent Tutoring Tools

A thesis presented in partial fulfilment of the requirements
for the degree of

Master of Information Systems

Department of Information Systems
Massey University
Palmerston North,
New Zealand

Jian Xu
2004

Abstract

In contrast with a traditional Intelligent Tutoring System (ITS), which attempts to be fairly comprehensive and covers enormous chunks of a discipline's subject matter, a basic Intelligent Tutoring Tool (ITT) (Patel & Kinshuk, 1997) has a narrow focus. It focuses on a single topic or a very small cluster of related topics. An ITT is regarded as a building block of a larger and more comprehensive tutoring system, which is fundamentally similar with the emerging technology "Learning Objects" (LOs) (LTSC, 2000a). While an individual ITT or LO focuses on a single topic or a very small cluster of knowledge, the importance of the automatic integration of interrelated ITTs or LOs is very clear. This integration can extend the scope of an individual ITT or LO, it can guide the user from a simple working model to a complex working model and provide the learner with a rich learning experience, which results in a higher level of learning.

This study reviews and analyses the Learning Objects technology, as well as its advantages and difficulties. Especially, the LOs integration mechanisms applied in the existing learning systems are discussed in detail. As a result, a new ITT integration framework is proposed which extends and formalizes the former ITT integration structures (Kinshuk & Patel, 1997, Kinshuk, et al. 2003) in two ways: identifying and organizing ITTs, and describing and networking ITTs. The proposed ITTs integration framework has the following four notions:

(1) Ontology, to set up an explicit conceptualisation in a particular domain,

(2) Object Design and Sequence Theory, to identify and arrange learning objects in a pedagogical way through the processes of decomposing principled skills, synthesising working models and placing these models on scales of increasing complexity,

(3) Metadata, to describe the identified ITTs and their interrelationships in a cross-platform XML format, and

(4) Integration Mechanism, to detect and activate the contextual relationship.

Acknowledgements

My thanks go first to my supervisor, Dr. Kinshuk for his guidance and help.

I also thank my family; they have supported me up with their love and patience throughout this research.

Finally, special thanks go to Marie for her reading and commenting.

TABLE OF CONTENTS

List of Figures

Chapter 2 Review of Learning Objects Technology

Chapter 3 ITT Integration Framework

Chapter 4 Toward the Prototype of Intelligent Tutoring Tools Integration System

List of Tables

Chapter 2 Review of Learning Objects Technology

Chapter 3 ITT Integration Framework

Chapter 4 Toward the Prototype of Intelligent Tutoring Tools Integration System

Chapter 1

Introduction

1.1 The Context of Traditional Intelligent Tutoring System

The notion of Intelligent Tutoring Systems (ITS) can be traced back to the late 1960's, when researchers began to explore the potential of "information structure-oriented" approaches to represent human cognition and learning (ADL, 2001a). At a very early stage, the fundamental requirements of ITS were outlined into three components (Shute & Psotka cited Hartley & Sleeman, 2002): (a) knowledge of the domain (expert model), (b) knowledge of the learner (student model), and (c) knowledge of teaching strategies (tutor model). The three components enable the system to assess what the student knows, to consider what the student needs to know and finally, to decide and present the curriculum elements following a teaching strategy. Furthermore, the application of artificial intelligence and cognitive psychology provided solutions to the problem of how to get a computer to behave intelligently (Shute & Psotka, 2002), which eventually led to the development of ITS.

Since then, researchers have addressed a number of important issues related to the ITS development, for example, in the 1970s, problem generation, simple student modeling, knowledge representation, and so forth; in the 1980s, model tracing, more buggy-based systems, case-based reasoning, and so forth; and in the 1990s, learner control, individual vs. collaborative learning, situated learning vs. information processing and so forth (Shute & Psotka, 2002).

However, Patel & Kinshuk (1997b) pointed out that the research in the field of Intelligent Tutoring Systems has failed to provide any substantial or viable systems that can be used in real academic environments due to the unclearly defined objectives and scopes of the system as well as the continuously shifting technological platform. They proposed a new approach to design and implement an Intelligent Tutoring System, namely the Intelligent Tutoring Tool (ITT).

1.2 The Context of Intelligent Tutoring Tools

A traditional Intelligent Tutoring System attempts to be fairly comprehensive and covers large chunks of a discipline's subject matter and provides tutoring facilities that try to satisfy all of the student, teacher, curriculum and institutional needs (Kinshuk, Shi & Patel, 2003). This approach proved to be an immense task and caused some problems, such as long development time, difficulty of maintenance, low reusability and so forth. ITTs were introduced as a solution by recongnising that curriculum needs to be broken down into small units which are covered by individual ITT. One of the advantages of this approach is that "the necessity of interpreting complex interactions and the construction of complex feedback messages is eliminated" (Patel & Kinshuk, 1997b). A student who has failed to grasp any of the intermediate steps is directed to an appropriate granularity level where simple feedback messages are adequate.

Apart from covering only a small domain content, the ITTs contain all the traditional components of ITS. They could be integrated into a network to provide higher-level intelligence and could be reused in other circumstances. Therefore, an ITT is viewed as a building block of the entire Intelligent Tutoring System, which has a clearly defined objective and covers a small amount of the domain content.

Similarly, an emerging learning content design, generation and implementation technology called "Learning Object" (LTSC, 2003), which is defined as "any entity, digital or non-digital, which can be used, re-used or referenced during technology-supported learning" (LOM, 2002), is envisaged as an element of computer-based instruction. A LEGO metaphor is frequently used to explain the basic idea of the Learning Object: small instructional elements (LEGOs) could be assembled into some larger instructional structures (like a castle and a robot) and the elements themselves could be reused in other instructional environments.

Based on the Learning Object definition, an ITT can be seen as a particular type of Learning Object. They share some fundamental similarities in their characteristics; however, ITT has some special aspects that are not applicable to other types of Learning Objects. For example, it is an intelligent entity, a self-contained object rather than a learning content in a basic form such as image, video, sound and text.

1.3 Motivation for the Research

While an individual ITT focuses on a single topic or a very small cluster of knowledge, automatic integration of interrelated ITTs is very important since it is believed to extend the scope of an individual ITT, guide the user from a simple working model to a complex working model and provide the learner with a rich learning experience, which results in a higher level intelligence.

Learning Object technology and its integration mechanisms applied in the existing learning system and standard have been explored in this project. However, a noticeable finding is that Learning Objects have to play a more important role in the integration process; that means that not all the Learning Objects such as learning contents in a basic form like image, sound and text could be used for the automatic integration process. The Learning Object should be an intelligent entity, which is a self-contained object with embedded functionalities, for example, the Intelligent Tutoring Tool type Learning Object.

Two types of integration have been proposed for the ITTs (Patel & Kinshuk, 1997b). These are (a) Vertical integration, a ranking ITT that allows holding and comparing results of different instances of an ITT, and (b) Horizontal integration, a Linking ITT that allows use of multiple tools to solve a given problem.

A prototype of a high-level Intelligent Tutoring Application through horizontal and vertical networking was developed successfully to exploit the two types of integration (Kinshuk, et al. 2003). In the prototype system, all the basic ITTs were networked or integrated via a high level Intelligent Tutoring Application. There were two important features implemented in the system. Firstly, the information for each basic Intelligent Tutoring Tool in the prototype system was stored in the database and was able to be accessed directly by the Intelligent Tutoring Application. Both the Intelligent Tutoring Application and Tool were designed and implemented independently, which sped up the development and made the maintenance easier. Secondly, a modular structure provided reusable functions to the Intelligent Tutoring Application. However, the prototype system did not provide a standard way to identify and organize independent ITTs or formalize the mechanism for ITTs' integration.

In terms of motivation, this research attempts to propose a framework that enables a formal integration of Intelligent Tutoring Tools. Furthermore, the context of ITTs are identified and organized in a pedagogical way, providing the learner a rich learning experience and guiding them from simple to complex concepts in order to achieve a higher level of learning.

1.4 The Research Steps

The project consists of the following five phases:

- ❖ Phase 1: Overviews the Learning Object and interrelated technology, metadata and architecture in general, discusses its concept, characteristics, standards, and application.
- ❖ Phase 2: Discusses the advantages and difficulties related to the Learning Object technology, investigates the integration mechanism applied in some existing learning systems and standards.
- ❖ Phase 3: Proposes the Intelligent Tutoring Tools integration framework.
- ❖ Phase 4: Implements a prototype system to examine the ideas proposed in the framework. The implementation also applies a prototype system development methodology
- ❖ Phase 5: Evaluates and assesses the prototype system.

1.5 Structure of the Thesis

The structure of the thesis closely follows the phases in the research steps. Chapter 2 contains a review of the Learning Object and related technology, followed by a discussion of the advantages and disadvantages of the emerging Learning Object technology. In Chapter 3, integration mechanisms applied in the existing learning systems and related problems are examined. As a result, a new integration framework is proposed and discussed in detail. In Chapter 4, after introducing a prototype development methodology, a prototype system is conducted through a complete system analysis, design and implementation process. Chapter 5 presents the evaluation of the prototype system, and Chapter 6 concludes the thesis by reviewing the work done in the project and discussing further research directions.

Chapter 2

Review of Learning Objects Technology

Hodgins (2000a) pointed out that common standards for Learning Objects, metadata, and learning architecture are mandatory for the success of the emerging Learning Objects technology. This chapter reviews the standard of these three collaborative components in the Learning Objects technology. The first section focuses on the Learning Objects (LOs) definition and characteristics. The second section reviews the metadata, focus on the Learning Objects Metadata (LOM), a standard used to describe learning resources; it also compares LOM with other common metadata standards. The third section explores some system architectures supporting the Learning Objects technology as well as the Shareable Content Object Reference Model (SCORM), a widely accepted e-learning industry standard. The fourth section presents a vision and considers the benefits of the new technology. The fifth section discusses some challenges and finally, a summary of the Learning Objects Technology is presented.

2.1 Learning Objects

The system design and implementation approach in traditional intelligent tutoring systems provided complete tutoring facilities to satisfy all the students, instructors, curriculum and institutional needs and requirements (Kinshuk, et al. 2003). However, this approach proved to be a difficult task that has caused some problems, such as long development time, difficulty of maintenance, low reusability and so forth. For example, in order to reuse the learning resources, Reigeluth and Nelson (as cited in Wiley, 2000) suggested that teachers have to (1) break the entire resources or courses down to constituent parts, (2) from these parts, they choose and reassemble them in different ways that support their individual instructional goals. To avoid this process and improve the efficiency of learning material generation, teachers should access the constituent parts, which have individual instructional objectives and cannot be broken down any further. The constituent part that covers a chunk of knowledge or concept is called a "Learning Object", which emerged as a solution to the problem, and a new technology to design, develop and deliver educational systems.

Firstly, the definition of Learning Objects is discussed in detail.

2.1.1 Definition

The IEEE Learning Technology Standards Committee (IEEE LTSC) defined Learning Objects as:

> Any entity, digital or non-digital, which can be used, re-used or referenced during technology-supported learning. Examples of technology-supported learning applications include computer-based training systems, interactive learning environments, intelligent computer-aided instruction systems, distance learning systems, web-based learning systems and collaborative learning environments.

> Examples of Learning Objects include multimedia content, instructional content, instructional software and software tools, referenced during technology supported learning. In a wider sense, Learning Objects could even include Learning Objectives, persons, organizations, or events (Wiley cited LOM, 2001).

However, this definition is too broad and unclear to be used in education content design and development. Besides the LTSC definition, a number of definitions have been offered by organizations and individuals. A detailed look at these definitions is given below.

Some researchers define Learning Objects in a narrower or refined way based on the LTSC definitions and try to set up models that are more practical and easier to implement. For example, in terms of the size or granularity of the Learning Objects, Shepherd (2000) pointed out that the Learning Objects in the LTSC definition could be as small as media assets, like images, paragraphs of text, audio clips etc. that could have less educational functionality, and as large as a fully self-contained piece of instruction, containing information, learning contents, mechanisms for practice and a means of assessment, that could be inflexible and hinder reusability etc. Hence, he defined "Learning Objects" between these two extremes as "A Learning Object is a small, reusable digital component that can be selectively applied – alone or in combination – by computer software, learning facilitators or learners themselves, to meet individual needs for learning or performance support." Another definition was from Wiley (2000), who identified some critical attributes of Learning Objects, i.e. reusability, digital resource and learning supportive, and defined “Learning Objects” explicitly and flexibly as “any digit resources that can be reused to support learning.” which was clearly a subset of the LTSC definition.

Others define their own type of Learning Objects to support the kinds of computer based training systems or e-learning solutions offered by their companies or implemented in their projects. For example, Cisco Systems, Inc., the networking hardware giant as well as e-learning solution provider, “recognizes a need to move from creating and delivering large inflexible training courses, to database-driven objects that can be reused, searched, and modified independent of their delivery media” (Cisco Systems, 2001). In their e-learning solution architecture, they introduced the following two terms: Reusable Learning Object (RLO) and Reusable Information Object (RIO), where RLO was described as a “lesson” built upon a Learning Objective, including overview, summary, assessment, and five to nine RIOs. RIO was described as a “section” with a simpler objective, which supported the RLO objective and had three items: content, practice and assessment. The structure of RIO and RLO is illustrated in Figure 2.1.

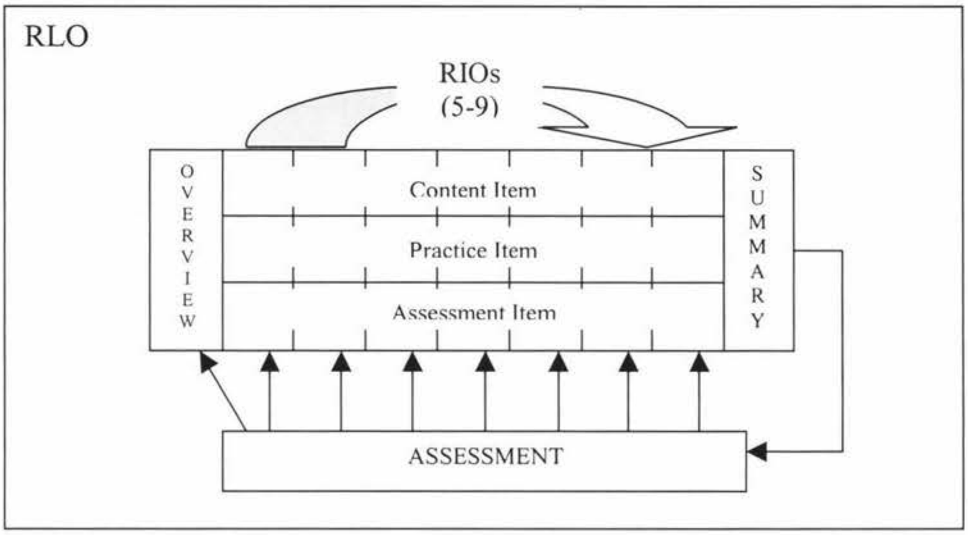


Figure 2.1. Reusable Learning Object (RLO) structure (Cisco Systems, 2001, p7)

IT e-learning provider NETg, Inc., used the term "NETg Learning Object (NLO)", in their skill builder structure, NLO was described as a topic (Learning Object structural component) which consists of three elements: (1) Learning Objective, (2) Learning Activity to teach the objective and (3) Assessment to determine if an objective has been met (L'Allier, 1998). Education Objects Economy, global communities for web based learning tools in JAVA (EOE, 2003), from the implementation technology's point of view, specifies that Learning Objects are downloadable and shareable instructional components in JAVA Applets, and has set up a Learning Objects library for free use.

Patel and Kinshuk (1997b) used "Intelligent Tutoring Tool" in their Byzantium project, where Intelligent Tutoring Tool was introduced as a building block of the entire Intelligent Tutoring System. However, unlike other types of Learning Objects, such as learning content in a basic form like image, sound, text and so forth, the ITT contained all the traditional components of ITS except for having a clearly defined objectives and covering only a small amount of the domain content. Not only were a knowledge base, student model and expert model covered in the ITT, a tutoring module, a user interface module, a level selector and a set of enhanced features which embedded functionalities such as a random question generator and dynamic feedback system were also provided. Therefore, the ITT was envisaged as "an autonomous entity possessing rudimentary intelligence" (Patel and Kinshuk, 1997b), an intelligent Learning Object.

Thirdly, synonymous terms for the "Learning Objects" are used more-or-less interchangeably. For example, Merrill (1998) introduced the "Knowledge Objects" as "a precise way to describe the subject matter, content or knowledge to be taught," and "a framework for identifying necessary knowledge components," and "a way to organize a data base of content resources so that a given instructional algorithm can be used to teach a variety of different contents". Advanced Distributed Learning (ADL) used the term "Shareable Content Object" (ADL, 2003b). Multimedia Educational Resource for Learning and Online Technology (MERLOT, 2003) referred to them as "online learning materials", while some other educational repositories providers, like Apple Learning Interchange (ALI, 2003) simply used "Learning Resources", TeleCampus (TeleCapmus, 2003), who list online courses from institutions throughout the world, used "courses" with different granularity levels (i.e. courses, modules and programs).

It seems that there are almost as many definitions of the "Learning Objects" as there are people employing them and it is likely that definitional debates and discussions will continue, the prevailing views (Warren, 2000, Wiley, 2000, Shepherd, 2000) suggest that the Learning Objects are an application of object-oriented thinking in the world of learning. The fundamental idea behind Learning Objects is that a relatively small instruction content can be built as a component that can be reused and repurposed in different learning environments. Based on this fundamental idea, the following discussion will take a detailed look at the characteristics of the Learning Objects.

2.1.2 Characteristics of the Learning Objects

The literature review suggests that Learning Objects have the following characteristics:

- Learning Objects are not the smallest components

  Learning Objects are combinations of a number of basic and individual media, such as images, sound, text, video clips, etc.

- Learning Objects are relatively small components

  Using conventional terminology, Learning Objects are considered as a concept or topic, which are constituent parts of a course, module, or lesson.

- Learning Objects are instructional components

  Learning is not "instructional theory neutral" (Wiley, 2000), instruction designers are required to provide an instruction objective and consistently apply design methodology for each Learning Object.

- Learning Objects are contextual independent

Learning Objects must not rely upon exposure to previous content in order to clearly provide instruction on a skill or concept. It is self-sufficient and can stand-alone.

❖ Learning Objects are content and presentation separated

Presentation itself consists of two parts; one is display format, such as font, background, layout, colours, etc. this information is usually stored in style sheets or templates. The other is the sequence or combination, in Learning Objects technology, metafiles are usually used to situate meaning and application and facilitate meaningful assembly.

❖ Learning Objects are searchable and retrievable

They are stored and accessed using standard metadata attributes and tags. With a search mechanism provided, the Learning Objects must be easy to identify for retrieval, access and evaluation. The details of Learning Object Metadata will be discussed in the next section - metadata.

❖ Learning Objects are shareable in a variety of Learning Management Systems (LMS)

One of the primary goals in the development of Learning Objects is to create learning contents that can run in different learning environments. The standardization of interface between Learning Objects and LMS, and the data model, is underway. Some details will be provided in the third section – infrastructure.

The above-mentioned characteristics of Learning Objects are illustrated in Figure 2.2.

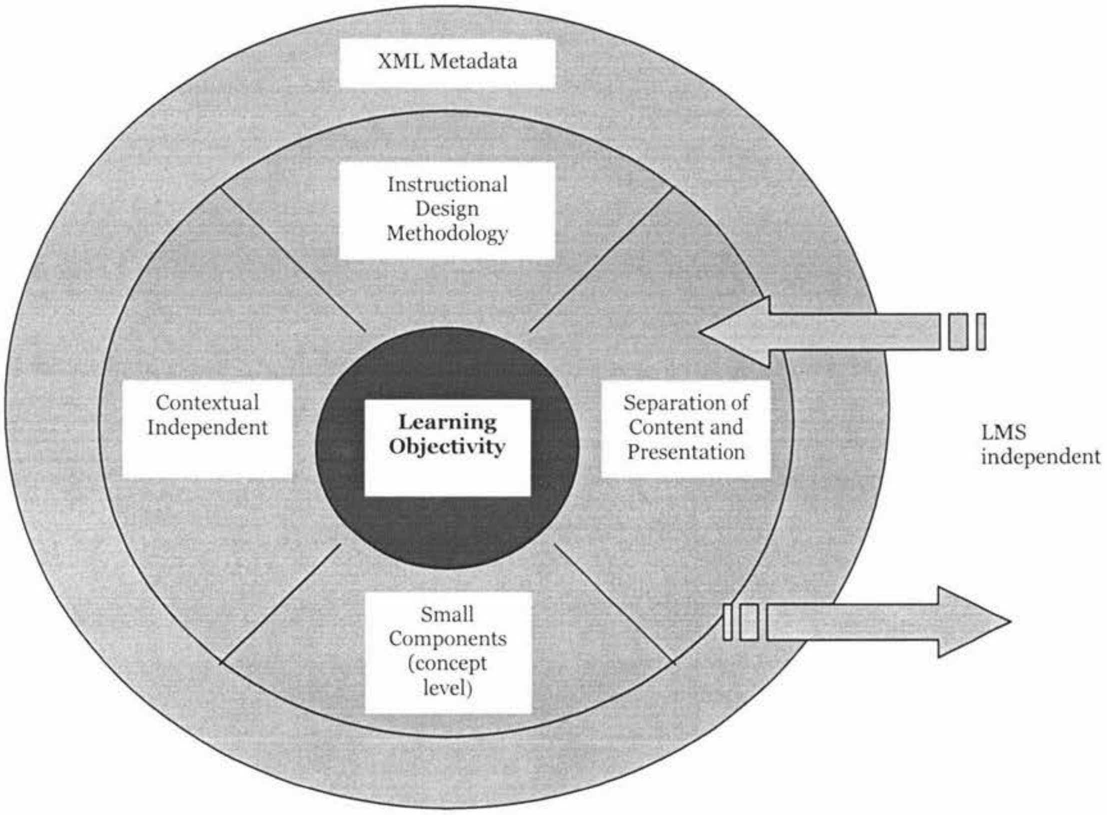

XML Metadata
Instructional Design Methodology
Contextual Independent
Learning Objectivity
Separation of Content and Presentation
Small Components (concept level)
LMS independent

Figure2.2. Graphic representation of the Learning Objects characteristics

2.2 Metadata

This section introduces the Metadata concept, followed by an introduction of the Dewey classification system as a classical example of metadata. It then discusses some widely accepted metadata standards for indexing, defining and searching education resources in detail; they are Dublin Core, Gateway to Educational Materials (GEM), Education Network Australia (EdNA) and Learning Object Metadata (LOM).

2.2.1 Metadata Concept

Metadata can be considered as "data about data". This is the historical definition of metadata. Thomas (Metadata guide, no date) suggested that metadata should be viewed from different points, which will help us to understand all parts of the concept. Metadata describes the characteristics of a resource; typically, a variety of descriptive information is presented: physical attributes (e.g. format, size), type (e.g. text or image, audio) and form (e.g. print copy, electronic file), as well as the originator of the resource, title and location. It uses a limited number of names which have meanings that the people within a domain agree to. It can be contained in a record separate from the resource, or bound with data in a structured format, and finally it provides sufficiency, scalability and interoperability to the resource.

Metadata shows many similar characteristics with the classification system that is used in libraries. This is introduced below as background knowledge of metadata.

2.2.2 Classification System in libraries

There are a number of classification systems used in libraries, for example, Sears is used mostly by secondary school libraries, and Library of Congress (LC) is used by most university libraries, Dewey Decimal Classification (DDC) is used by public libraries and some secondary schools. Among these classification systems, Dewey Decimal Classification (DDC) system is the world's most widely used library classification system.

The Dewey Decimal Classification (DDC) system was developed by Melivil Dewey in 1876, and has since been greatly modified and expanded to keep pace with knowledge and enhance the efficiency and accuracy of the classification work. Online Computer Library Centre (OCLC) introduces the DDC system as an ideal general knowledge organization tool, and highlights classification and notation as the key mechanisms used in the DDC system:

> Classification provides a system for organizing knowledge. Classification may be used to organize knowledge represented in any form, e.g., books, documents, electronic resources.
>
> Notation is the system of symbols used to represent the classes in a classification system. In the Dewey Decimal Classification, the notation is expressed in Arabic numerals. The notation gives both the unique meaning of the class and its relation to other classes. The notation provides a universal language to identify the class and related classes, regardless of the fact that different words or languages may be used to describe the class. (OCLC, 2003)

The literature review (OCLC, 2003 and NNDSB, 2003) shows that DDC organizes the entire world knowledge into ten different broad disciplines or fields of study, called main classes, numbered 000 - 999. The ten main classes are:

000	Generalities
100	Philosophy & Psychology
200	Religion
300	Social Sciences
400	Language
500	Natural Sciences & Mathematics
600	Technology (Applied Science)
700	Arts & Recreation
800	Literature
900	History & Geography

Materials that are too general to belong to a specific group (e.g., encyclopaedias, newspapers, magazines, general periodicals, etc.) are placed in the 000's. This class is also used for certain disciplines that deal with knowledge and information (e.g., computer science, internet, library and information science, journalism and publishing & news media). Each of the other main classes comprises a major discipline, or group of related disciplines. Each main class is further divided into ten subclasses and the number is then divided up into 10's, for example, the social science subclasses are:

300	Sociology and Anthropology
310	General Statistics

320	Political Science
330	Economics
340	Law
350	Public Administration
360	Social Services; Associations
370	Education
380	Commerce, Communications, Transport
390	Customs, Etiquette, Folklore

The next level of categorization is delineated by 1's, for example; the next level categories of education are:

371.1 Teaching and Teaching Personnel

371.2 School Administration and Management

371.3 Methods of Instruction and Study

371.4 Guidance and Counselling

371.5 School Discipline

371.6 Physical Plant

371.7 School Health and Safety

371.8 The Student

371.9 Special Education

Smaller divisions to subdivide the category even further are shown by adding decimals. The classification and notation mechanism is presented in Table 2.1.

In summary, the DDC classification schema (1) describes the whole world knowledge based on well-defined categories; it is widely accepted and agreed to by the people from libraries in the world. The Dewey Classification Editorial Policy Committee (EPC), an international board whose members are from public, special, and academic libraries as well as library schools, reviews the proposed revisions and expansions from the editorial office, and advises the maintenance and development of the classification system. (2) The DDC Classification schema is presented in a well-developed structural hierarchy. An important concept called "hierarchical force" is the principal of the structural hierarchy, it enforces that all topics (aside from the ten main classes) are part of all the broader topics above them and the corollary is also true, that is whatever is true of the whole is true of the parts (OCLC, 2003). It also builds up network relationships (i.e. super ordinate, subordinate and coordinate) between the topics. (3) The DDC Classification schema provides a semantic notation mechanism in a universally recognized format, Arabic numerals. It provides efficiency, accessibility and scalability to the access of resource. The metadata for Electronic Resources is discussed below.

Table2.1 Dewey Decimal Classification (DDC) System

Main Class	SubClasses	Divisions	Smaller Divisions
000 Generalities			
100 Philosophy & Psychology			
200 Religion			
300 Social Sciences	300 Sociology and anthropology 310 General statistics 320 Political science 330 Economics 340 Law 350 Public administration 360 Social services; associations 370 Education 380 Commerce, communications, transport 390 Customs, etiquette, folklore	370 Education 371 School management; special education 372 Elementary education 373 Secondary education 374 Adult education 375 Curriculum 376 Education of women 377 Schools & religion 378 Higher education 379 Government regulation, control, support	371.1 Teaching and teaching personnel 371.2 School administration and management 371.3 Methods of instruction and study 371.4 Guidance and counselling 371.5 School discipline 371.6 Physical plant 371.7 School health and safety 371.8 The student 371.9 Special education
400 Language			
:::			
800 Literature & Rhetoric			
900 Geography & History			

2.2.3 Dublin Core

The Dublin Core (DC) metadata standard, developed and maintained by the Dublin Core Metadata Initiative (DCMI), is "a simple yet effective element set for describing a wide range of networked resources" (DCMI, 2003).

The DC metadata standard includes two levels: simple and qualified. The former comprises fifteen basic elements; the latter enhances the simple version with one more element, "Audience", and a group of element refinements that make the meaning of an element more specific, as well as some encoding schema that aid in the interpretation of an element value. For example, a Date element has a list of refinements, Created/Valid/Available/Issued/Modified/Copyrighted/Submitted, to specify the event's date in the life-cycle of a resource, meanwhile, it also has two encoding schema, W3C-DTF and DCMI Period to interpret the date format. Those elements can be grouped into three descriptive categories (DCMI, 2003):

❖ ***Content***:

- ✓ Title - Name of the resource
- ✓ Subject - Topic of the content of the resource
- ✓ Description - Account of the content of the resource
- ✓ Type - Nature or genre of the content of the resource
- ✓ Source - Reference to a resource from which the present resource is derived
- ✓ Relation - Reference to a related resource

- ✓ Coverage - Extent or scope of the content of the resource
- ✓ Audience - Class of entity for whom the resource is intended or useful

❖ ***Intellectual Property***:

- ✓ Creator - Entity primarily responsible for making the content of the resource
- ✓ Publisher - Entity responsible for making the resource available
- ✓ Contributor – Entity responsible for making contributions to the content of the resource
- ✓ Rights - Information about rights held in and over the resource

❖ ***Instantiation***:

- ✓ Date - Date associated with an event in the life cycle of the resource
- ✓ Format - Physical or digital manifestation of the resource
- ✓ Identifer - Unambiguous reference to the resource within a given context
- ✓ Language - Language of the intellectual content of the resource

2.2.3.1 Design Principles

The mission of DCMI is to make it easier to discover and retrieve networked information via a simple, widespread metadata standard. An international, cross-disciplinary group of professionals in DCMI has been working on this mission, which designs the DC Metadata as a simple, easy-to-understand, cross-subjects, and extensible standard. The following gives some more details about the design principles of the DC metadata.

Simplicity and Easy-To-Understand

The DC element set has been designed small and simple with easy-to-understand definitions. It helps a wide range of people to apply the metadata easily and cost-effectively to their resources description.

Interoperability and Cross-Disciplinary

The DC metadata has been designed to support a common or generic set of elements from one field of knowledge to another; it is widely used across disciplines and communities of practice. The wide adoption of DC metadata by resource description communities, such as museums, libraries, publishers, government agencies, international organisations and commercial organizations has proved its effectiveness across different subjects.

Modularity

The DC metadata supports metadata modularity, which means multiple, disparate, complementary metadata instances combined together for a single resource description (Sutton & Mason, 2001). Resources Description Framework (RDF) is a part of the infrastructure to support the metadata modularity. The principal of the concept has been expressed in "Application Profiles", which "consists of data elements drawn from one or more namespace schemas combined together by implementers and optimised for a particular local application." (Heery & Patel, 2000)

Extensibility

As mentioned previously, the DC metadata has added elements, element refinements as well as an encoding schema to the DC simple version to define an element in a more specific and meaningful way. It also allows implementers to extend or create additional elements and element qualifiers where necessary to meet the specialized needs of local or domain specific applications.

2.2.3.2 Summary

In summary, the DC metadata has benefited from its simple, cross-disciplinary, modularity and extensibility design principles and achieved wide acceptance. However, DCMI also realizes that the Dublin Core Metadata Elements Sets (DCMES) is insufficient in some domain-specific applications so it has established subject-specific working groups, for example, government, libraries and education, to explore the possibility of new elements and qualifies generation based on the DCMES.

Based on Dublin Core metadata standard, some educational metadata standards have been developed, which are introduced below.

2.2.4 Gateway to Educational Materials (GEM)

The Gateway to Educational Materials (GEM) project, sponsored by the U.S. Department of Education, is to provide educators an efficient and effective way to find educational resources that are distributed on web sites across the Internet (GEM, 2003a). The key to the project's success is the establishment of metadata standard, "a Gateway", to quality collections of education resources.

2.2.4.1 Design Principles

The GEM metadata standard is based on the Dublin Core metadata standard. The GEM has extended DC metadata in the following two ways. (1) Additional elements are added to meet the needs of educational resource discovery and retrieval. In addition to the fifteen DC elements, GEM adds seven elements (GEM, 2003b), which are summarised in table 2.2. (2) The elements are refined through a number of GEM controlled vocabularies as well as encoding schemes. As an example, the values used to describe "GEM.Pedagogy.Grouping" from GEM Pedagogy Element Controlled Vocabulary (GEM, 2003b) have been presented in table 2.3.

Table 2.2

GEM 2.0 Elements and Semantics (Extension from DC only)

Element	Definition
Audience	A class of entity for whom the resource is intended.

	(* Audience is an element in the DC qualified version)
Cataloguing	Information about the individual and/or agency that created the GEM catalogue record.
Duration	The recommended time or number of sessions needed to effectively use the resource being described.
Essential Resources	A brief free-text listing of materials essential to the successful use of the entity by the teacher as stated in the entity being described.
Pedagogy	Pedagogical methods and procedures, which also can be used to specify student instructional groupings, teaching methods as well as assessment methods etc.
Quality [Deprecated]	Assessment of the quality of the resource being described.
Standards	State and/or national academic standards mapped to the entity being described.

Table 2.3

GEM 2.0 Controlled Vocabularies (GEM.Pedagogy.Grouping)

Element.Refinements (Controlled Values)	**Pedagogy.Grouping (Description)**
Cross age teaching	Utilization of older students from higher grade levels to provide increased help and attention for younger students at lower grade levels. (ERIC)
Homogeneous grouping	Organization or classification of students according to specified criteria for the purpose of forming instructional groups with a high degree of similarity. (ERIC)
Individualized instruction	Adapting instruction to individual needs within the group. (note: do not confuse with "independent study" or "individual instruction") (ERIC)
Large Group instruction	Teaching of students in large classroom situations. (note: do not

	confuse with "mass instruction") (ERIC)
Non-graded instructional grouping	Grouping students according to such characteristics as academic achievement, mental and physical ability, or emotional development rather than by age or grade level. (ERIC)
Small group instruction	Form of teaching possible in higher education (and sixth forms) where the student to staff ratio is relatively low, and allowing for intensive interaction among small group of participants. (DOE)

2.2.4.2 Summary

The GEM extends Dublin Core element sets, a widely implemented, easy-to-extend and cross-disciplinary standard, with some education-specific elements as well as qualifiers. However, GEM "relies heavily on the use of element qualification while holding the number of new, domain specific elements to a minimum" (Sutton & Mason, 2001). It has inherited some of the advantages of DC metadata as a simple and easy to use standard.

2.2.5 Education Network Australia (EdNA)

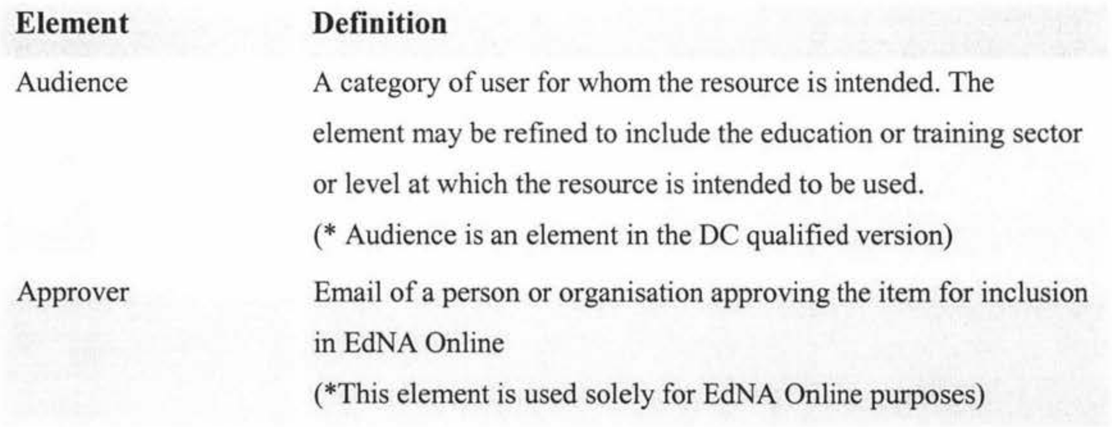
The Education Network Australia (EdNA) Online is a service that supports and promotes the benefits of the Internet for learning, education and training in Australia, it is managed by education.au limited, a non-profit company limited by guarantee and owned by the Australian education and training Ministers (EdNA, 2003). "EdNA Online is one of the largest repositories of education and training resources in Australia and relies on metadata for its resource discovery and retrieval services. One of the major functions is creating, maintaining and extending EdNA metadata" (Millea, 2003).

2.2.5.1 Design Principles

"The EdNA metadata standard is based on the internationally recognised Dublin Core metadata element set (DCMES) and is consistent with the Australian Government Locator Service (AGLS)" (EdNA, 2003). AGLS Metadata Set itself is a standard based on Dublin Core, which aims to describe Australian government resources and enables them to interchange and access resources across government departments. Similar to GEM's extension from DC metadata, EdNA creates new elements, element refinements and encoding schemes to describe education-relevant information about resources or for management purpose. Table 2.4 summaries the EdNA's element set extended from DC. (EdNA, 2003), Table 2.5 shows an example of the element refinements and encoding scheme for EdNA.Audience.Sector (EdNA, 2003).

Table 2.4.

EdNA v1.1 Elements and Semantics (Extension from DC only)



Element	Definition
Audience	A category of user for whom the resource is intended. The element may be refined to include the education or training sector or level at which the resource is intended to be used. (* Audience is an element in the DC qualified version)
Approver	Email of a person or organisation approving the item for inclusion in EdNA Online (*This element is used solely for EdNA Online purposes)

CategoryCode	A numerical code derived from the database tables which support the EdNA Online Browse Categories (*This element is used solely for EdNA Online purposes)
Entered	Data item was entered as an entry in the online item database (*used for management purpose and for EdNA Online purposes)
Indexing	To what extent should EdNA Online indexing ("spidering") software follow links from this page (*This element is used solely for EdNA Online purposes)
Review	A third party review of the resource. This element is defined such that two forms of review are accommodated – a short evaluative comment or a more formal review pointed to by a URL.
Reviewer	Name of person and /or organisation or authority affiliated with the review
Version	Version of the EdNA metadata standard applied

Table 2.5

EdNA 1.1 Controlled Vocabularies (EdNA.Audience.Sector)

Element.Refinements	**Audience.Sector**
(Controlled Values)	**(Description)**
ACE	Adult and community education
Higher Education	
Preschool Education	
School Education	
VET	Vocational Education and Training

2.2.5.2 Summary

Similar to GEM, EdNA extends Dublin Core with new element sets as well as qualifiers, some of the new added elements/qualifiers in EdNA are educational-relevant, some of them however are added particularly for the EdNA Online, or for the management purposes. The GEM and EdNA's "different development paths and modelling choices" result in little interoperability between the extensions of these two DC-Based educational metadata (Sutton & Mason, 2001).

Like GEM and EdNA, many metadata schemas are DCMI-based, however, several projects, for example, CanCore, ARIADNE and SCORM are more related or rooted in the emerging Learning Object Metadata (LOM) standard from The IEEE Learning Technology Standards Committee (LTSC). A detailed look at LOM is presented below.

2.2.6 IEEE LTSC Learning Object Metadata (LOM)

The IEEE (Institute of Electrical and Electronics Engineers) Learning Technology Standards Committee (LTSC) is chartered by the IEEE Computer Society Standards Activity Board to develop accredited technical standards, recommended practices, and guides for learning technology. The Learning Object Metadata (LOM) working group under LTSC has created the LOM metadata standard, which originated from a European project ARIADNE and a United State project IMS, and it has also referred to Dublin Core work. The purpose of the LOM Standard is to "facilitate search, evaluation, acquisition, and use of Learning Objects, ...(it) also facilitates the sharing and exchange of Learning Objects" (LTSC, 2003).

2.2.6.1 Design Principles

The LOM metadata schema is used to describe characteristics of a Learning Object. In the LOM standard, those characteristics are grouped into the following nine categories:

1: The *General* category provides the general information of a Learning Object such as title, language, keywords and so forth.

2: The *Life Cycle* category groups the history and current state of the learning resource as well as the contributors, for example, version and status.

3: The *Meta-Metadata* category provides information about the metadata standard used, including metadata identifier, contributors and so forth.

4: The *Technica*l category describes technical characteristics and requirements of the Learning Object, such as format, size, hardware and software required to use the resource, some other platform requirements and so forth.

5: The *Educational* category groups the educational or pedagogic characteristics of this resource, for example, the interactivity type, the audiences for which this resource is designed and so forth.

6: The *Rights* category provides all the information about rights, conditions and the cost for using this resource.

7: The *Relation* category defines the relationships between Learning Objects. The relationship definitions are based on Dublin Core.

8: The *Annotation* category provides some comments from those who have used the Learning Object in an educational context, as well as the user information.

9: The *Classification* category describes the particular classification system that the learning resource is applied to.

The entire LOM metadata schema is defined as a hierarchical model or a tree structure. At the top of the hierarchy is the "root" element. The root element contains the above-mentioned nine categorized elements, which groups many sub-elements. A sub-element is called a "branch" if it contains sub-elements; otherwise, it is called "leaves". The tree structure is illustrated in Figure 2.3 using the sample elements from IEEE LTSC LOM Draft Standard.

In terms of data elements, a "branch" element is also called an aggregate data element while a "leaf" element is called a simple data element. For the entire data element, LOM schema defines its name, explanation, size and order, as well as an example, for simple data element. It also specifies its value space as well as the data type.

2.2.6.2 Summary

In contrast to the Dublin Core Metadata Standard that defines a minimum number of cross-disciplinary elements, enhanced with elements refinements and encoding schema, LOM has defined a hierarchical conceptual data schema dedicated to the

Learning Objects. That means the data elements sets are well defined to describe relevant characteristics of the Learning Objects and ensure a high degree of semantic interoperability in the education domain.

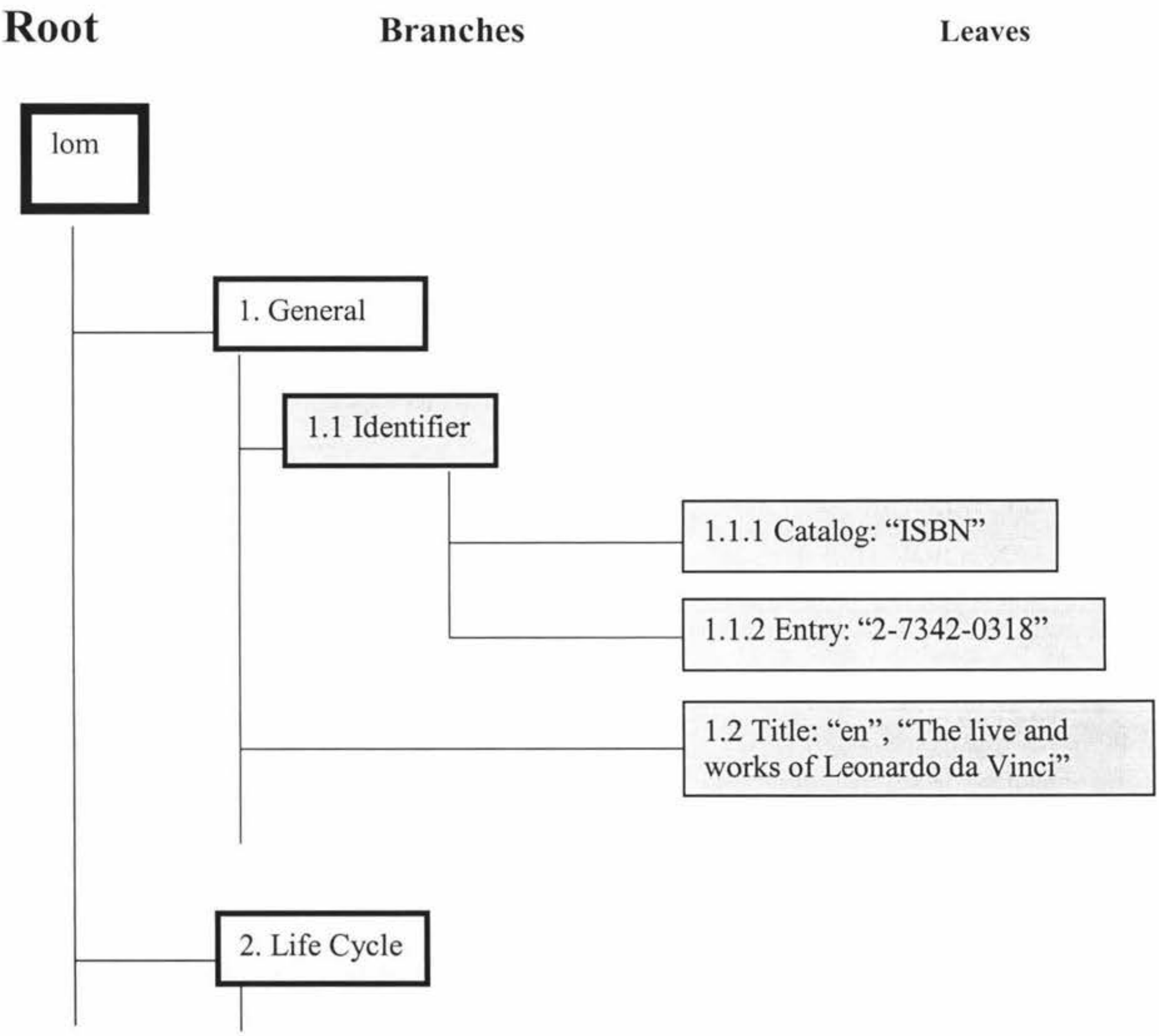

Root
Branches
Leaves
lom
1. General
1.1 Identifier
1.1.1 Catalog: "ISBN"
1.1.2 Entry: "2-7342-0318"
1.2 Title: "en", "The live and works of Leonardo da Vinci"
2. Life Cycle

Figure 2.3. Hierarchical view of Learning Object metadata elements

2.2.7 Metadata Mapping

The metadata schemes basically aim to provide a standard for resource discovery and retrieval and increase the reusability and value of the resources. However, a large number of metadata standards have been developed and the variety of metadata standards makes the interoperability of metadata difficult. To reach a wide acceptance, metadata must be made available in accordance with some popular metadata standards. That means a metadata created and maintained in one standard should be able to be accessed by another related standard. Mapping one metadata to another is an attempt to make metadata interoperable. This section presents the mapping between the metadata introduced above.

LOM (2002) provides mapping to the unqualified Dublin Core metadata standard, which is illustrated in Table 2.6. According to the draft standard, a further refinement of the mapping to the qualified Dublin Core metadata can be achieved.

Table 2.6.

Mapping to Unqualified Dublin Core (LOM, 2002. Table B.1 – Mapping to Unqualified Dublin Core)

DUBLIN CORE ELEMENT	LOM ELEMENT
DC.Identifier	1.1.2:General.Identifier.Entry
DC.Title	1.2:General.Title

DC.Language	1.3:General.Language
DC.Description	1.4:General.Description
DC.Subject	1.5:General.Keywords or 9: Classification with 9.1:Classification.Purpose equals “Discipline” or “Idea”.
DC.Coverage	1.6:General.Coverage
DC.Type	5.2:Educational.LearningResourceType
DC.Date	2.3.3:Life cycle.Contribute.Date when 2.3.1:Life cycle.Contribute.Role has a value of "Publisher".
DC.Creator	2.3.2:Life cycle.Contribute.Entity when 2.3.1:Life cycle.Contribute.Role has a value of "Author".
DC.OtherContributor	2.3.2:Life cycle.Contribute.Entity with the type of contribution specified in 2.3.1:Life cycle.Contribute.Role.
DC.Publisher	2.3.2:Life cycle.Contribute.Entity when 2.3.1:Life cycle.Contribute.Role has a value of "Publisher".
DC.Format	4.1:Technical.Format
DC.Rights	6.3:Rights.Description
DC.Relation	7.2.2:Relation.Resource.Description
DC.Source	7.2:Relation.Resource when the value of 7.1:Relation.Kind is "IsBasedOn".

GEM and EdNA are DCMI-based metadata standard. Gong (2002) provided mappings from GEM and EdNA to LOM, which are illustrated in the Table 2.7 and Table 2.8.

Table 2.7.

GEM mapping to LOM (Gong, 2002)

GEM METADATA ELEMENTS	LOM'S METADATA ELEMENTS
GEM. Audience	5.5:Educational.Intended end user role
GEM. Cataloguing	9.2.1:Classification.Source
GEM. Duration	5.9:Educational.Typical Learning Time
GEM. Grade	5.6:Educational.Context & 5.7:Educational.Typical Age Range
GEM. Pedagogy	9.1:Classification.Purpose & 5.2: Educational. Learning Resources Type
GEM. Quality	4:Technical
GEM. Standards	2.1:Life cycle.Version

Table 2.8

EdNA mapping to LOM (Gong, 2002)

EdNA Metadata Elements	LOM's Metadata Elements
EDNA. Audience	5.5:Educational.Intended end user role
EDNA. Approver	8.1:Annotation.Person
EDNA. CategoryCode	9.2.2.1:Classification.Id
EDNA. Version:	2.1:Life cycle.Version
EDNA. Review	8.3:Annotation.Description, the content of this annotation
EDNA. Reviewer	8.1:Annotation.Person, the person or organization who created this annotation.

2.3 Architecture

With a large number of Learning Objects developed, and the standardization of learning metadata, supportive architectures or backbones of the e-learning systems are highly demanded. This section introduces some architecture models that are widely accepted in the education system implementation, providing a framework to deploy Learning Objects technology, and taking full advantage of the above-introduced Learning Objects concepts and IEEE Learning Objects Metadata. These architecture models are grouped or levelled as follows:

- Learning Objects Repository (LOR)
- Learning Management System (LMS)
- Learning Content Management System (LCMS)

2.3.1 Learning Object Repository

A Learning Object Repository (LOR) is a simple, and easily implemented, architecture. A LOR typically allows learners, educators or researchers to search and retrieve either Learning Objects or Universal Resource Locator (URL) and to access the required Learning Objects from the repository. Technically, a LOR consists of presentation component that is typically a web-based user interface, and a database component that contains collections of Learning Objects or references to the Learning Objects, as well as metadata. (Duncan, 2001)

2.3.1.1 Functions

Currently, Learning Objects Repositories have different focal points and differ in the type and complexity of the functions they offer. Duncan summarized a typical use of a Learning Objects Repository as: (1) locate, (2) preview, (3) borrow and (4) publish. (Duncan, 2001)

❖ Locate

Browsing and Searching are the two methods to locate required Learning Objects. **Browsing** allows the user to go through the learning contents based on well-defined categories, either subjects or metadata elements/qualifiers such as resource type, media type and so forth. For example, Multimedia Educational Resource for Learning and Online Teaching (MERLOT) allows the user to browse the collection by subjects, such as arts, business, education, humanities, mathematics, science & technology and social science (MERLOT, 2003). Whilst iLumina, a cooperative project funded by the national science foundation, allows the user to find resources not only from disciplines, but also by resource type (e.g. lesson, exercise, example), structure (e.g. collection or individual learning resource), media type (e.g. Image, portable document, video, audio, web page) as well as contributors (e.g. publisher, submitter, author) (iLumina, 2003). The browsing function is helpful for a new user of the system to get a quick impression of the contents. **Searching** is another powerful method providing an efficient and accurate way to retrieve resources. Searching criteria are based on metadata elements or qualifiers. For example, the advanced search function provided in MERLOT allows the user to enter data about twenty specific fields such as subject category, material type, title, description and so forth. EdNA, however, provides standard, advanced and metadata search functions. The advanced search function provides interaction with other LOR's such as GEM, MERLOT and VOCED and the metadata search function allows the user to specify a combined search based on metadata elements (EdNA, 2003).

- **Preview**

Once the Learning Objects have been located, contents, quality and cost, as well as level of the learning resource, need to be evaluated to make sure it serves the intended learning or teaching objectives. While downloading and examining the Learning Objects by the end user is the most obvious and straightforward approach, some LORs, like MERLOT also provide detailed peer reviews to help the user to make right choices.

- **Borrow**

In most of the Learning Objects repositories such as MERLOT, EdNA, iLumina and so forth, the end users are allowed to download the learning resources for free. However, some learning resources are chargeable, and some resources are restricted to members or a certain group of users.

- **Publish**

Anyone with a proper authority (in the case of MERLOT, a registered member) can be a publisher or an author. He/she is encouraged to upload and publish his/her work into the repositories and the work is then shared and reused by the repository user. However, association of the metadata with Learning Objects is done either by the contributor through a metadata template provided, or by the LORs 'librarian'.

2.3.1.2 Architecture of Learning Objects Repository

As introduced above, Learning Objects Repositories consist of an interface component, typically a web interface, and a database component that contains Learning Objects or a reference to the Learning Objects and metadata. Apart from these, the architecture of the Learning Object repository can be categorized as a client-server system and peer-to-peer network (Neven & Duval, 2002). The former is easy to implement and maintain due to the centralized Learning Objects and metadata while the latter has the advantages of ease of setup and scalability. Currently, most of the Learning Objects Repositories are implemented in Client/Server architecture. The architecture of a typical centralized C/S Learning Objects Repository is illustrated in Figure 2.4.

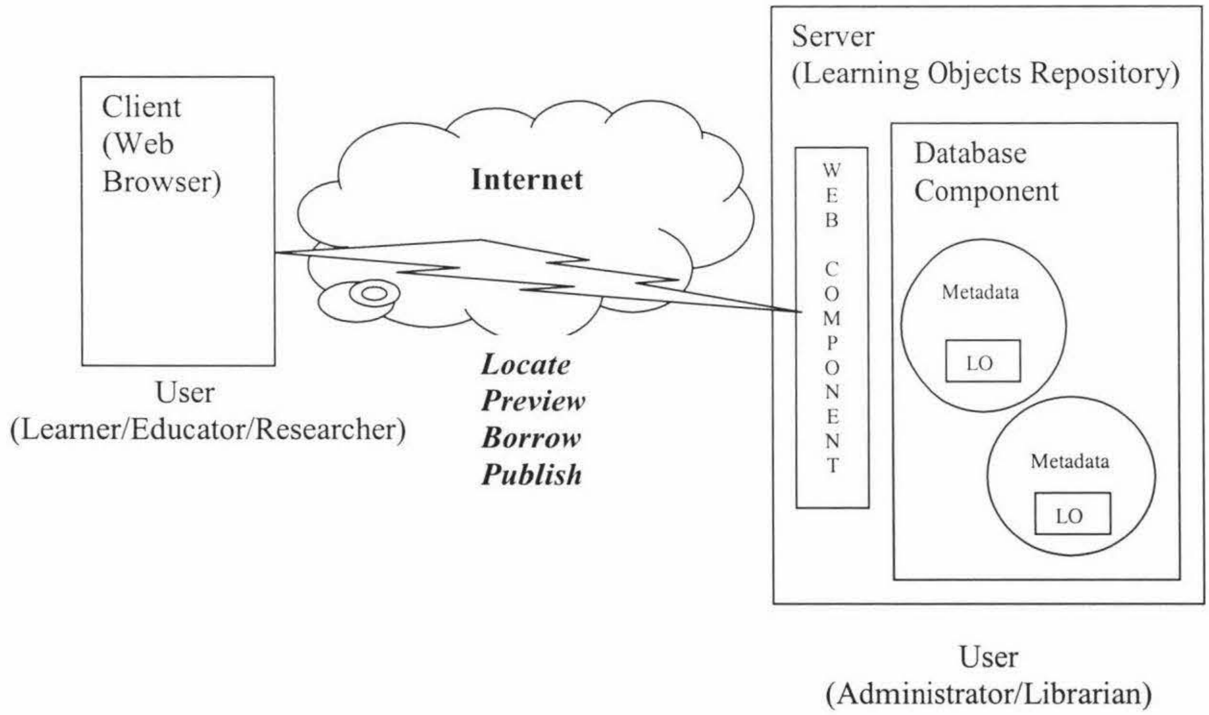


Figure 2.4. Architecture of a typical centralized Learning Objects repository

2.3.2 LMS and LCMS

The terms of Learning Management System (LMS) and Learning Content Management System (LCMS) have been used interchangeably among e-learning users and buyers since they confuse the similar features and functions provided by the two systems. In fact, these two systems have different focuses and strengths; they are distinct and complementary with some shared components. It is easier to understand the concepts by comparing and explaining the two systems together rather than discussing them separately.

According to Learning Circuits, an online magazine all about e-learning from the American Society for Training & Development (ASTD),

A Learning Management System (LMS) is defined as:

> Software that automates the administration of training. The LMS registers users, tracks courses in a catalogue, records data from learners; and provides reports to management. An LMS is typically designed to handle courses by multiple publishers and providers. It usually doesn't include its own authoring capabilities; instead, it focuses on managing courses created by a variety of other sources (ASTD, 2003).

A Learning Content Management System (LCMS) is defined as:

> A software application (or set of applications) that manages the creation, storage, use, and reuse of learning content. LCMSs often store content in granular forms such as Learning Objects (ASTD, 2003).

2.3.2.1 Comparisons between LMS & LCMS

The definitions of LMS and LCMS show that Learning Management System and Learning Content Management System have distinct focuses and strengths. Duncan (as cited in Ellis, 2001) explained the basic difference between LMSs and LCMSs as "An LMS solves (problems in) running a learning organization and an LCMS gets the right content to the right people at the right time." The definitions also describe features and functions provided by the two systems, however, not all the LMS and LCMS have the some features and functions. The vendors of these systems differentiate their products with some unique features and functions. Despite these differences, the LMS and LCMS have their own basic components. LMS consists of five basic components (Ielearning, 2003). The system components are:

- ❖ The **courseware launching** component that sequences instructional activities for a student, and provides the interface for student access to the activities. (This element of an LMS is commonly referred to as "Computer Managed Instruction" or "CMI").

- The **course-development** component enables a course administrator to specify the content of the course in terms of lessons and the sequence of these lessons.

- The **roster operations** component registers the student and enrols him or her in courses.

- The **assignment management** component assigns the lessons to the student and records the student performance data.

- The **data collection** component provides the automated collection and management of data.

LCMS also shares some system components, according to IDC (Brennan, Funke & Anderson, 2001). The system components are:

- Authoring Tools: Help knowledge experts with little or no programming experience to create new or reuse existing learning contents in multiple formats, like HTML, Word, PowerPoint, Flash and so forth.
- Learning Objects Repository: A central place to store and manage Learning Objects, the Learning Objects may be delivered individually, or used as a component of a larger learning module or course.
- Dynamic Delivery Interface: Dynamically and adaptively deliver learning contents via the Web, CD-ROM, or printed materials based on the learner profile, protests and/or user queries.

- ❖ Administration Application: Manage student records, launch courses, as well as track student progress.

Based on the literature review (Greenberg, 2002, Donello, 2002, Jacobsen, 2002, Rengarajan, 2001, Brennan, et al., 2001 & Ellis, 2001), detailed comparisons between Learning Management System and Learning Content Management System focus on the following areas, roles, functions, target user, management focus, content tracking level, creation capability, standard, delivery and administration, the differences are summarized in Table 2.9.

Table 2.9

Differences Between LMS And LCMS

	LMS	LCMS
Primary Role	Manage learning activities and competencies	Manage Learning Objects and serve up to the right learner at the right time
Major Functions	Catalogue the learning content Schedule and register students for courses Launch courses Keep track learner progress	Create, store, reuse, manage, personalize, deliver and improve learning content
Primary Target Users	Training managers, instructors, administrators	Content developers, instructional designers, project managers
Management focus	Learners	Contents or Learning Objects
Content Tracking level	Higher-level (Course)	Lower-level (Learning Object)
Content creation capabilities	No	Provide author tools which help creating new or reuse

		existing Learning Objects
Content Standard	No	Based on reusable Learning Objects
Content Delivery	Deliver prescribed, off-the-shelf or slightly customized courses	Provide user interface to navigate content Deliver adaptive content in multiple formats
Administration (Data)	A rich learner profile, including organizational affiliations, job role, preferences, competencies, skill levels and so forth Course	Learning Objects only (stored in a Learning Object repository)
Administration (Report)	Primary focus on managing reports for training results and competency mapping, skill gap analysis	Secondary focus
Interface to other System	Yes	No

2.3.2.2 Integration of LMS & LCMS

The Learning Management System and Learning Content Management System have fundamentally different focuses and strengths. However, they also provide complementary or partially overlapped functionalities in the following areas (Rengarajan, 2001):

- **Content:**

The LMS focuses on course-level management, it manages, delivers, and tracks courses, which are typically composed of Learning Objects that were created, managed and tracked in LCMS.

- **Users:**

The LMS focuses on the learners in the whole organization, it helps the users reduce their skill gaps and improve the individual and organizational competency via managing current competency status, analyzing skill gaps and recommending suitable courses. Therefore, the LMS typically maintains a rich user profile. One of the goals of the LCMS is to deliver the right learning content to the right people, the personalized learning content delivery is mainly achieved via the productivity and reusability of learning content.

- **Administration:**

From the administration viewpoint, the LMS focuses on the learner management with a high-level (course) content tracking. In contrast, the LCMS pays more attention to the learning content management and enhancement, as well as the interaction between user and Learning Objects rather than the management of users themselves.

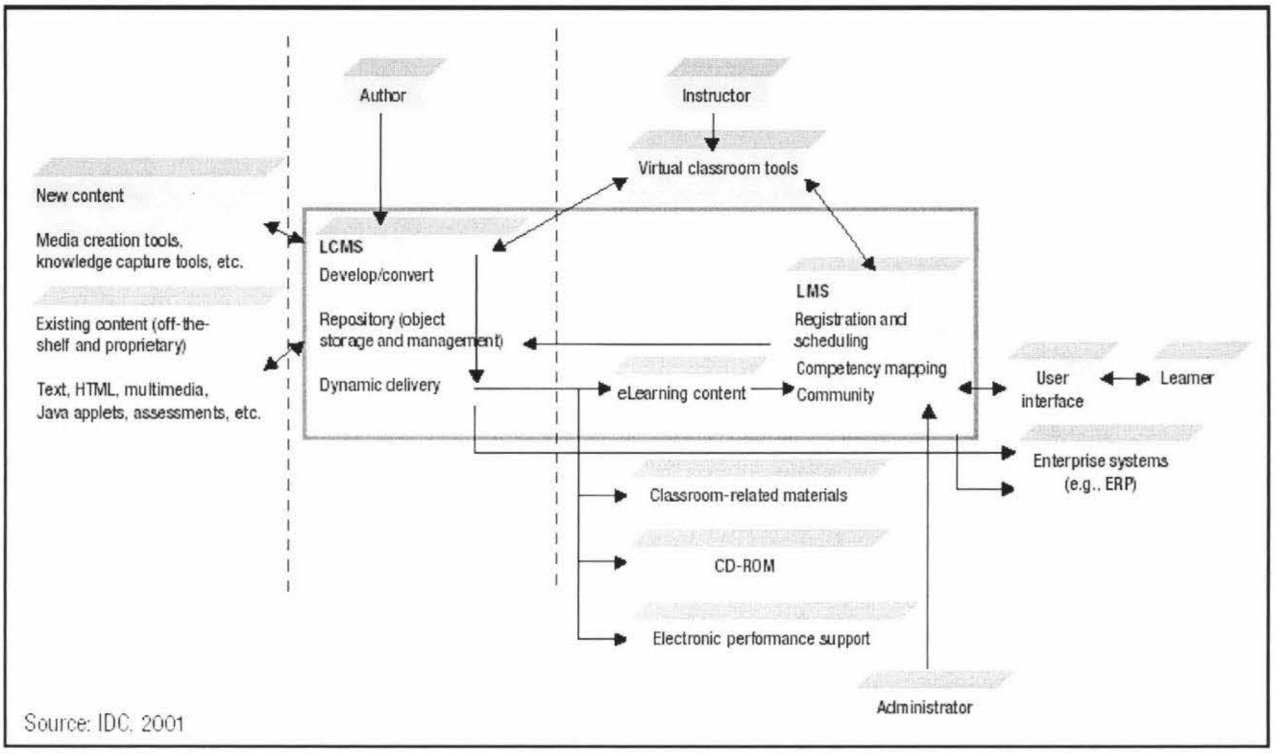
The complementary focuses, strengths and interests of these two systems in contents, users and administration result in a high demand for the integration of LMS and LCMS. The tight integration of these two systems can ensure a richer learning experience for the user and a more comprehensive tool for learning environment administration. As Brennan, et al. (2001) stated:

> An LMS can manage communities of users, allowing each of them to launch the appropriate objects stored and managed by the LCMS. In delivering the content, the LCMS also bookmarks the individual learning learner's progress, records test scores, and passes them back to the LMS for reporting purposes. (Brennan, et al., 2001)

The integration of LMS and LCMS is highlighted and illustrated in Figure 2.5 from "The Learning Content Management System" (Brennan, et al., 2001), which offers the following key benefits as Rengarajan (2001) stated:

- Advanced personalized learning content based on the richer user profiles
- Shared and Reusable Learning Objects
- Improvement of the learning content
- Intangible knowledge capturing
- Integrated security
- Unified administration and maintenance
- Uniform search capabilities

However, the key to integration success is an open, interoperable approach that is based on common standards. The following section takes a detailed look at one standard widely accepted in the e-learning industry, the Shareable Content Objects Reference Model (SCORM).


Author
Instructor
Virtual classroom tools
New content
Media creation tools, knowledge capture tools, etc.
Existing content (off-the-shelf and proprietary)
Text, HTML, multimedia, Java applets, assessments, etc.
LCMS
Develop/convert
Repository (object storage and management)
Dynamic delivery
LMS
Registration and scheduling
Competency mapping
Community
eLearning content
User interface
Learner
Enterprise systems (e.g., ERP)
Classroom-related materials
CD-ROM
Electronic performance support
Administrator

Source: IDC, 2001

Figure 2.5. LMS and LCMS integration in an e-learning environment (Brennan, et al., 2001)

2.3.3 Shareable Content Objects Reference Model (SCORM)

The Advanced Distributed Learning (ADL, 2003a) initiative was established by the United States Department of Defense (DoD) in 1997 with a goal to "develop a strategy for applying learning and information technologies to modernize education and training and to promote cooperation between government, academia and business to develop e-learning standardization" (ADL, 2001a). The Shareable Content Objects Reference Model (SCORM) is a product of the efforts.

SCORM is a set of specifications, originated in other organizations including IMS Global Learning Consortium, Inc., the Aviation Industry CBT (Computer-Based Training) Committee (AICC), the Alliance of Remote and Instructional Authoring & Distribution Networks for Europe (ARIADNE) and the Institute of Electrical and Electronics Engineers (IEEE) Learning Technology Standards Committee (LTSC), for developing, packaging, accessing and delivering high-quality education and training materials that are tailored to individual needs whenever and wherever they are required (ADL, 2001a) . SCORM v1.3 basically has four parts, Overview, Content Aggregation Model (CAM), Run Time Environment (RTE) and Sequence and Navigation (SN):

- **Overview**: contains the history, vision and future of ADL and SCORM as well as a summary of the technical specifications and guidelines in other parts.
- **Content Aggregation Model (CAM)**: contains guidance for index learning resources and puts them into a structured learning content so it can be moved and reused in different learning system.
- **Run Time Environment (RTE)**: contains guidance for Application Program Interface (API) to launch content, and data model to track and report the learner's progress.
- **Sequencing and Navigation (SN)**: contains guidance to represent complex sequencing of the content object through a set of learner-initiated or system-initiated navigation events.

A detailed introduction of CAM, RTE and SN follows.

2.3.3.1 CAM

The SCORM Content Aggregation Model (CAM) provides guidance for assembling, labelling and packaging content. CAM focuses on the content model components, content packages and metadata. In CAM, there are two types of learning resources:

- Assets: the learning content in a basic form such as text, image, sound, web page and so forth.
- Shareable Content Objects (SCOs): A collection of one or more assets that include a specific asset that can be launched to communicate with the LMS. It is also the lowest level of granularity of content that can be tracked by a SCORM-conformant learning system.

Figure 2.6 illustrates the concepts of assets and SCOs.

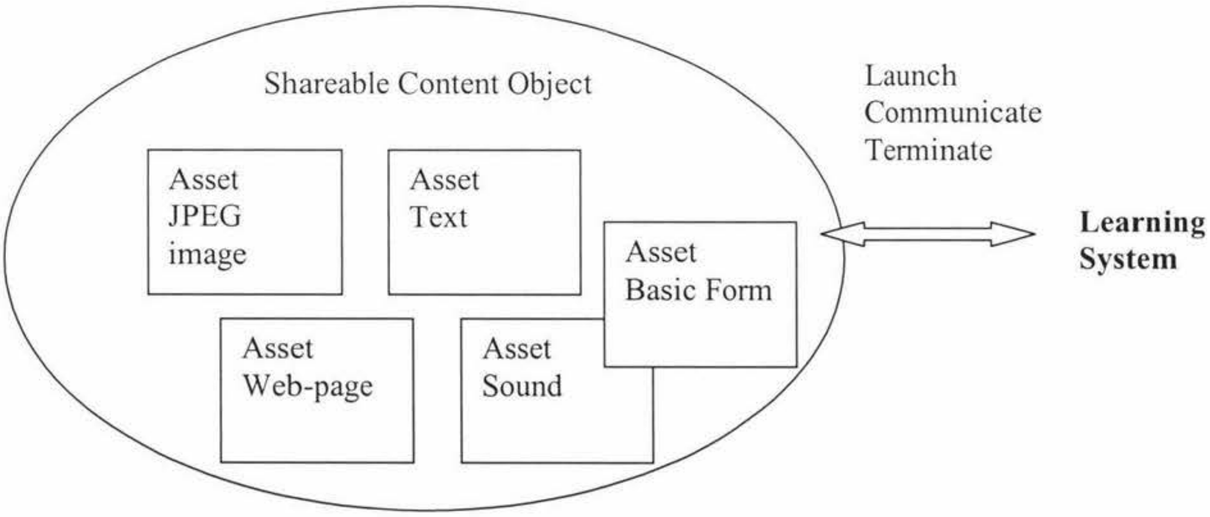


Figure 2.6. Assets and shareable content object in SCORM

To provide a particular learning experience, these learning resources (assets and SCOs) are aggregated and sequenced on two different levels:

- Activity: A unit of instruction.
- Content Aggregation: A content structure that describes cohesive activities, and associates learning taxonomies to the activities, the specification is derived from AICC.

The concepts are illustrated in Figure 2.7, and the whole set is considered as a package, a unit of learning. The standard content packaging specification is from IMS, which provides a standardized way to exchange packages between systems.

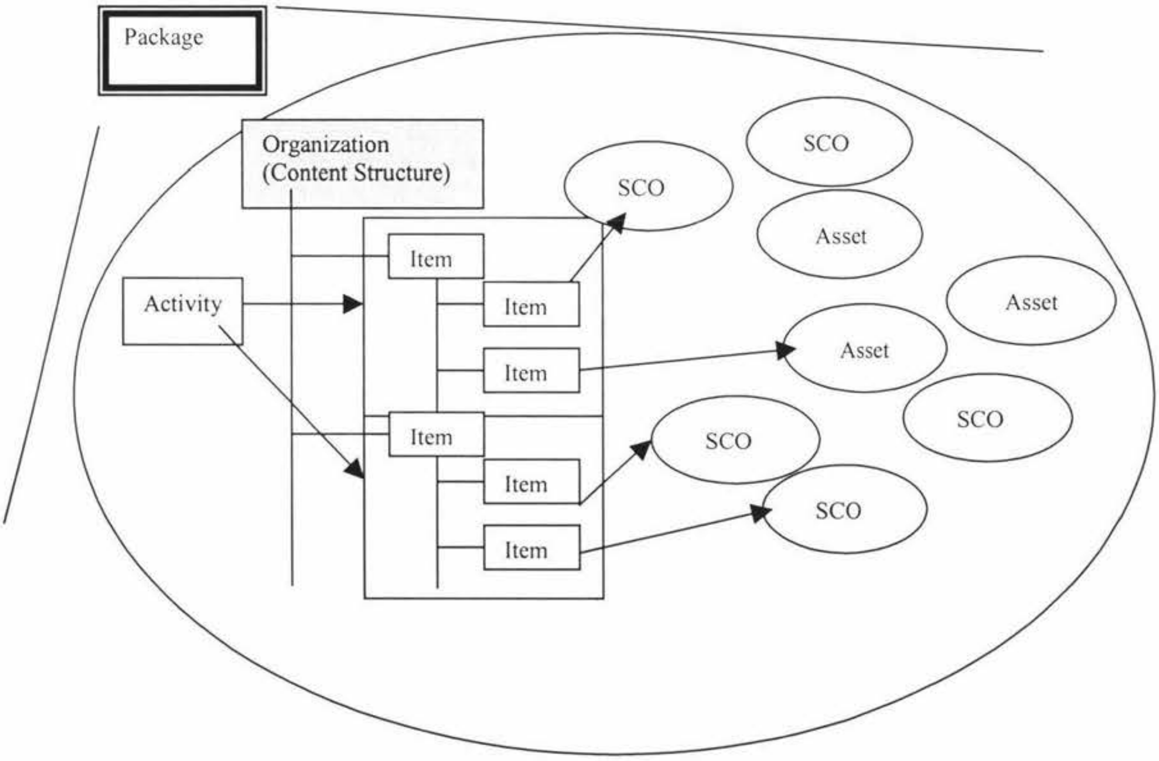


Figure 2.7. Content packaging in SCORM

Finally, CAM metadata, based on the IEEE LTSC LOM, describes the SCORM content model components in each level, these are package, content aggregation, activity, SCO and Assets. It provides content reusability and accessibility at various levels. The XML "binding" for the metadata specification is from IMS. The relationship of the content package's manifest file and the content model components is illustrated in Figure 2.8.

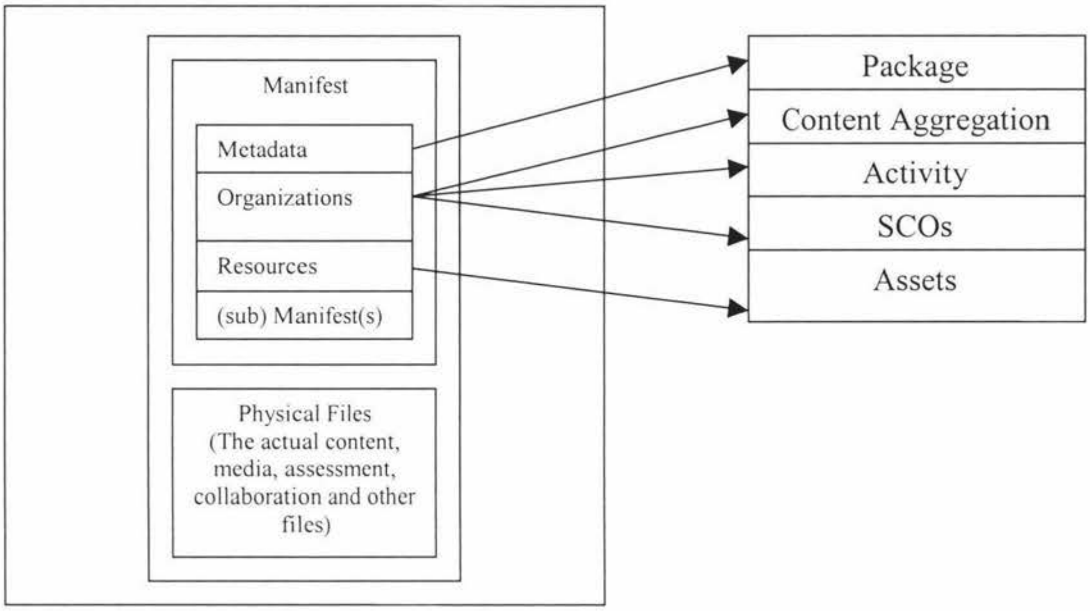


Figure 2.8. Metadata corresponding to content model components

2.3.3.2 RTE

"Two goals of the SCORM are that content objects be reusable and interoperable across multiple LMSs" (ADL, 2003c). For this to be possible, the SCORM Run-Time Environment (RTE) provides:

- A common way to start Content Objects, either Assets or SCOs.
- A common mechanism for SCOs to communicate with LMS,

❖ A common data set to track the learner's experience.

As illustrated in Figure 2.9, the three aspects of the Run-Time Environment are Launch, Application Program Interface (API) and Data Model.

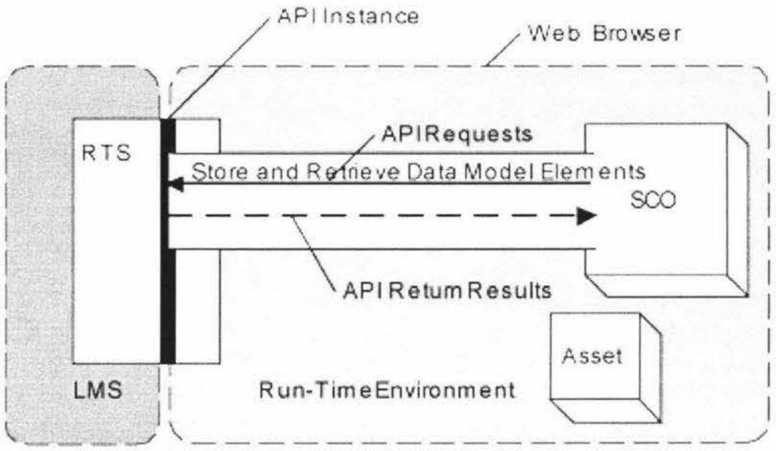


Figure 2.9: SCORM conceptual run-time environment. (ADL, 2003c)

The Launch mechanism defines a common way for LMSs to start Web-based learning resources, either SCOs or Assets. A content object is launched based on either a pre-defined sequence in the content structure, or the learner's performance in previous learning experiences. The system also can process and deliver the content object based on the users' requirements. The communication protocols are standardized through the use of a common API.

The API provides the communication mechanism for informing the system of the state of the content objects. In SCORM, the content object is SCO, since only SCO communicates with LMS at run-time. API also provides methods to get and set data (e.g. score and time in content) between the LMS and the Content Object. The SCORM API is based directly on the run-time environment functionality defined in AICC's CMI001 Guidelines for Interoperability document.

A Data Model is a standard data set used to define the information being communicated. The information specified what both the LMS and SCO are expected to know, such as the completion status, comments of the user and credit. A standardized data model not only standardizes how the system tracks learners, but also ensures data interchangeability among multiple LMSs. The SCORM data model derived directly from the AICC EMI Data Model described in the AICC CMI Guidelines for Interoperability.

2.3.3.3 SN

The SCORM Sequencing and Navigation (SN) is partially based on the IMS Simple Sequencing (SS) Specification Version 1.0, which defines "a method for representing the intended behaviour of an authored learning experience such that any LMS will sequence discrete learning activities in a consistent way" (ADL, 2003d). SCORM SN applies and extends the IMS SS Specification to describe the sequence of learning activities, and dynamically evaluate and identify the next presenting activities based on the learner-content object interaction and navigation patterns. For this to be possible, SN provides standard specifications on (ADL, 2003d):

❖ Activity Tree

Activity Tree defines a tree structure of learning activities that are roughly defined as a meaningful unit of instruction. Activity Tree consists of (1) cluster, which includes a single parent learning activity, and its immediate children activities, (2) leaf activity, which is the end of the activity tree without any child activity.

- Data Model

The data model is a set of pre-defined data elements to define a method for expressing rules, events and conditions as well as run-time behaviours associated with various sequencing and navigation. Typically, there are three different types of data model: (1) Dynamic run-time Tracking Model for maintaining the state of individual learning activity as well as capturing learner-content object interaction information, (2) Dynamic run-time Activity State Model for maintaining the global state of the whole activity tree, and (3) Static Sequencing Definition Model defining authored sequencing intentions, which are described in the content package. For example, the "Sequencing Control Choice" element in the Sequencing Definition Model indicates that in a cluster, a learner is able to choose any child-learning activity if the parent activity is "sequencing control choice" true.

- Sequencing Process

"The overall sequencing process provides the overarching control process for the LMS's sequencing implementation" (ADL, 2003d). The various sequencing behaviours encapsulated in the overall sequencing process are defined in the SCORM SN. They are:

Navigation Behaviour – Describes how a navigation request is validated and translated into termination and sequencing requests.

Termination Behaviour – Describes how the current attempt on an activity ends, how the state of the activity tree is updated, and if some action should be performed due to the attempt ending.

Rollup Behaviour – Describes how tracking information for cluster activities is derived from the tracking information of its child activities.

Selection and Randomization Behaviour – Describes how the activities in a cluster should be considered during processing a sequencing request.

Sequencing Behaviour – Describes how a sequencing request is processed on an Activity Tree in attempt to identify the "next" activity to deliver.

Delivery Behaviour – Describes how an activity identified for delivery is validated for delivery, and how an LMS should handle delivery of a validated activity. (ADL, 2003d)

❖ Navigation Model

The Navigation Model defines a set of navigation events that can be triggered by LMS or SCO. It also defines the corresponding navigation request for each event.

2.3.3.4 Summary

As described above, CAM, RTE and SN cover different concepts and key technologies in the e-learning open architecture. They also show some relationships with each other. Table 2.10, mainly from ADL 2003b, shows the concepts and key technologies covered in each part and the relationship between them; the organizations and the originated source specifications are added.

Table 2.10

SCORM Coverage (ADL 2003b)

SCORM	Concepts Covered	Key SCORM Technology Covered	Areas of Overlap	Source
Overview	High-level Conceptual information	Incidental mention of numerous elements of SCORM terminology	Covers areas of the SCORM RTE, CAM and SN books at a high-level	
Content Aggregation Model (CAM)	Assembling, labelling and packaging content	SCO, Asset, Content Aggregation, Package, Package Interchange File (PIF), Metadata, Manifest, Sequencing Information, Navigation Information	SCOs and manifests. SCOs communicate with an LMS via the RTE. Manifests contain Sequencing and Navigation information	IEEE LTSC - LOM, IMS - Content Packaging IMS - XML Binding AICC – Content Structure
Run-Time Environment	LMS management of the run-time	API, API instance, Launch, Session	SCOs which are covered in the	AICC - CMI data model

(RTE)	environment which includes launch, content to LMS communication, tracking, data transfer, error handling	Methods, Data Transfer methods, Support Methods, Temporal Model, Run-time data model	SCORM CAM book, are content objects which use the RTE	AICC - CMI Guidelines for interoperability
Sequencing and Navigation (SN)	Sequencing Content, navigation	Activity Tree, Learning Activities, Sequencing Information, Navigation information, navigation data model	Sequencing and Navigation affects how content is assembled in a manifest	IMS – Simple Sequencing

2.4. Vision and Benefits

Hodgins (2000b) presented a vision for the Learning Object technology, stating that the new technology "represents a completely new conceptual model for the mass of content used in the context of learning." It will forever change the way that people capture knowledge, deliver learning contents and manage resources, and eventually improve human learning and performance.

The Learning Objects, which are broken down into small chunks of information and labelled with standardized metadata descriptions, will enable users to effectively discover, reuse or repurpose them. Meanwhile, with these knowledge building blocks, the automatic assembly of personalized learning content based on the user profile can be easily created and delivered to best satisfy the individual needs. Furthermore, not only can the Learning Objects be put into a centralized server base system which allows users to access and manage resources, but also a real-time search, peer-based decentralized system will have more impact on learning, learners and learning content. The decentralized system will connect Learning Objects together; those connected Learning Objects are able to communicate, pass data, manipulate the other and cooperate together to provide higher-level intelligence. Eventually, the Learning Objects technology will provide a personalized, dynamic and adaptive learning environment, that will allow the user to learn from anywhere at anytime, which is believed, in the knowledge-based economy, will lead to high performances for an individual and an organisation. (Hodgins, 2000a, 2000b)

The benefits of the Learning Objects are well summarized with a list of "abilities":

- Accessibility: access instructional components from one remote location and deliver them to many other locations
- Interoperability: use instructional components developed in one location with one set of tools or platform in another location with a different set of tools or platform
- Adaptability: tailor instruction to individual and situational needs
- Reusability: incorporate instructional components into multiple applications

- Durability: operate instructional components when base technology changes, without redesign or recoding
- Affordability: increase learning effectiveness significantly while reducing time and costs (Hodgins, 2000a cited Parmentier, 1999)

Based on the bright future and benefits that the Learning Object technology promised, government and industry have put their efforts into theoretical research and practical implementation of the new technology. However, the lack of successful educational systems which apply Learning Object technology in the market shows a number of problems and challenges the Learning Object technology is facing. These are discussed in the next section.

2.5 Challenges

Learning object and interrelated metadata and learning architecture have become the buzzwords in the e-learning industry, as they are expected to be an evolutional solution to the learning contents design, development, delivery, and be able to provide "anywhere, anytime" learning environments. However, the application of this new technology has shown a number of problems and challenges, which have been addressed by Friesen (2003) and Wiley (2003), as discussed below.

1. Granularity

In the Learning Object definition, the granularity, the size of the Learning Object, has not been clearly defined. It can be as large as a whole course, obviously hard to reuse in another educational context. It also can be as small as a picture, text statement or an animation. Wiley (2000) saw this problem from two perspectives.

> From an "efficiency" point of view, the decision regarding Learning Object granularity can be viewed as a trade-off between the possible benefits of reuse and the expense of cataloguing.
>
> From an instructional point of view, alternatively, the decision between how much or how little to include in Learning Object can be viewed as a problem of "scope".

In the same work, Wiley also pointed out that reusability is the core of Learning Object technology, and the decision about "scope" depends on the learning situation and applied instruction design methodology.

2. Learning in the e-learning standards

Learning Objects and the standards are expected to be able to support multiple learning environments and instruction methodologies, accordingly, they are frequently described as "pedagogically neutral". However, Friesen (2003) points out that "specifications and applications that are truly pedagogically neutral cannot also be pedagogically relevant". Which means it is necessary to connect instruction design methodology to Learning Object design as Wiley (2001) advocated.

3. Decontextualized design and contextualized learning

Wiley (2003) pointed out that "the instructional design behind Learning Objects is increasingly moving toward decontextualization. This is true because of an inversely proportional relationship between the size of a Learning Object and its potential for reuse." However, from his previous research, he stated that "Learning Object 'use' is better described as 'contextualization'". (2003)

Patel and Kinshuk (1997) cited Brezillon and Abu-Hakima's finding that "knowledge had a contextual component and that the context provided a principled way to cluster, partition and organize knowledge". This finding also explains that a simple linkage between the decontextualized Learning Objects is insufficient to provide a meaningful context for learning (Wiley, 2003).

4. Automation

Learning Object technology is also expected to be able to provide personalized lessons through a computer-automated assembly. However, so far there has not been any significant work done in this automating assembly process. That is largely because "Automating these processes is also a knowledge intensive activity likely to require the application of Artificial Intelligence techniques such as knowledge representation and reasoning" (Mohan & Brooks, 2003).

2.6. Summary

Firstly, this chapter reviewed Learning Object technology in general, an emerging instruction design and implementation technology. Learning Object technology consists of three interrelated components: (1) Learning Object is a self-contained, reusable instructional component, (2) Metadata is standardised descriptive data used to describe the Learning Object, which enables effective and efficient object retrieval and interchange, (3) Learning Architecture is a backbone system, which controls, manages and delivers learning contents for individual needs. Secondly, it presented the vision and benefits of the emerging Learning Object technology, and finally some critical problems and challenges were discussed.

The next chapter will have a detailed look at the Learning Objects' integration approaches in existing learning systems, then point out that in a particular learning scenario, a new integration framework is necessary. Finally, an integration framework is proposed.

Chapter 3

Integration Framework

Chapter 2 has reviewed the standards for Learning Objects, metadata and learning architecture. The emergence of the new instruction design and implementation technology shows a flexible, efficient and effective way to generate reusable education resources that will significantly change the way that people capture knowledge, deliver learning contents and manage resources, as well as improve human learning and performance. Although there are some critical problems and challenges with the new Learning Objects technology, the new object-oriented approach in learning content generation and delivery has a bright future and offers significant benefits compared with traditional ITS approach.

While an individual LO focuses on a single topic or a very small cluster of knowledge, the importance of the automatic integration of interrelated LOs is very clear. It is believed to extend the scope of an individual LO and help the learner to learn and master complex skills efficiently, which results in a higher level of learning. This chapter overviews some learning systems focusing on their Learning Objects integration approaches. Secondly, a circumstance or a situation in which the necessity of a new Integration Framework is described. Finally, the characteristics of the framework are proposed and discussed in detail.

3.1 Learning Systems Overview

This section reviews some learning systems focusing on their approaches to assembling, sequencing and representing Learning Objects.

3.1.1 SCORM

In the SCORM, Learning Objects are assembled, labelled and packaged into the content package with a prescribed content structure (ADL, 2003b). A SCORM Content Package may include additional information that describes how an LMS is intended to initialize, process the Content Package, and manage its contents. It may also include information about the content structure, which is eventually converted into an activity tree. The activity tree is defined in the SCORM Sequencing and Navigation as "a conceptual structure of learning activities managed by the LMS for each learner" (ADL, 2003b). It describes a structure or sequence of learning activities used by LMS to present the learning contents. Meanwhile, LMS is able to adjust the pre-defined sequence based on the learner's progress and performance.

In SCORM, it is also emphasised that it is the responsibility of LMS to launch a Learning Object. Any Learning Object is not allowed to launch other Learning Objects, because "if a learning resource contained a 'hardwired' branching to another learning resource under specific conditions, it could not be used in a different course in which the second learning resource might not be applicable or available" (ADL, 2003b).

The mechanism applied in SCORM is illustrated in Figure 3.1.

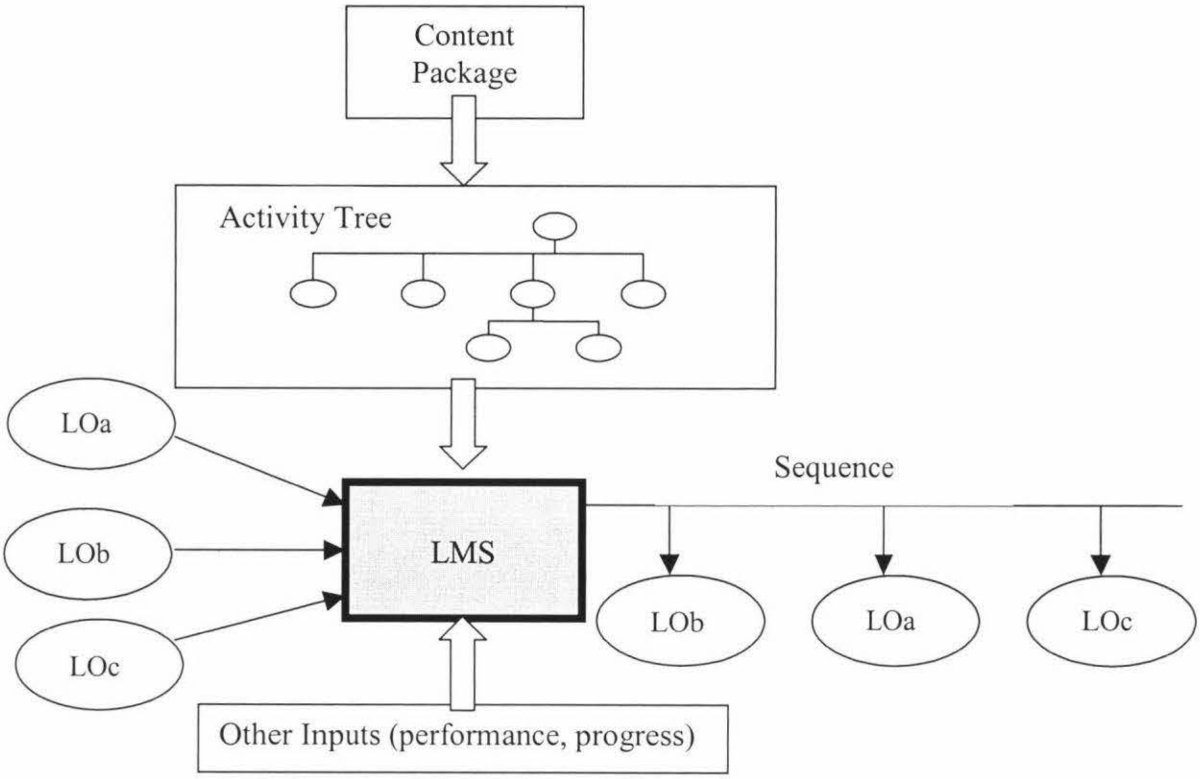


Figure 3.1. Sequence Mechanisms in SCORM

In summary, the approach for sequencing Learning Objects in SCORM is: (1) based on Content Aggregation, which contains the content structure that is converted into activities trees for sequence purpose, (2) it largely depends on the possibilities of the Learning Management System, which tracks the progress and performance of learners and has the responsibility to launch a learning resource, (3) it does not allow a Learning Object to launch another Learning Object.

3.1.2 Multimedia Learning Objects (MLO) Project

Bradley and Boyle's Multimedia Learning Objects project was intended to help with the teaching of introductory java programming in the Department of Computing at London Metropolitan University. In order to provide pedagogically rich experiences with the cohesive and simple Learning Object, Boyle (2003) chose to create compound Learning Objects to provide multiple perspectives of a given learning topic. "The compound object consists of two or more independent objects that are linked to create the compound: a base object and one or more optional expansion objects" (Bradley & Boyle, 2003). Figure 3.2 shows the design structure; a text-based example serves as the base object, then in the link column links are provided to the expansion objects, usually multimedia objects such as animation and code examples.

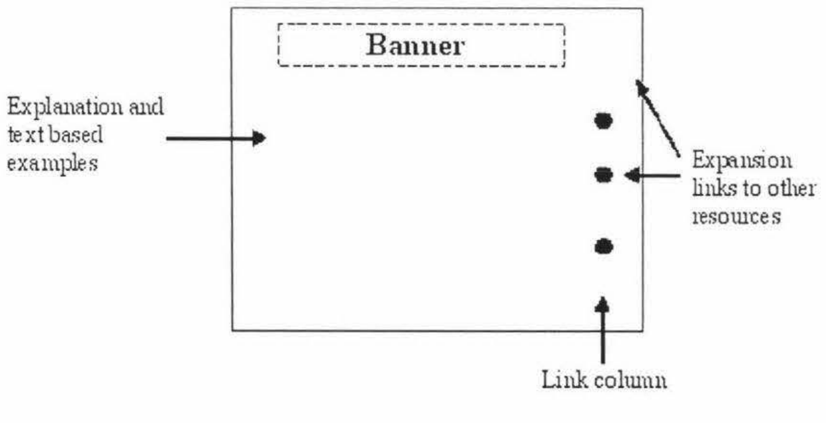

Banner
Explanation and text based examples
Expansion links to other resources
Link column

Figure 3.2: Schematic layout of format for Learning Object realisation (Boyle, 2003)

Boyle (2003) highlighted that this structure "provided a basis for repurposing through the addition or deletion of objects to amplify or shape the pedagogical richness of the compound object." A pre-defined compound is presented in default, but the instructors are allowed to reconfigure this to their own compound object.

The compound object provides an alternative view for a single learning topic through linking expansion objects with the base object. However, the whole course structure and sequence of Learning Objects is presented through a syllabus, which is pre-defined. Figure 3.3 shows the design structure of a syllabus.

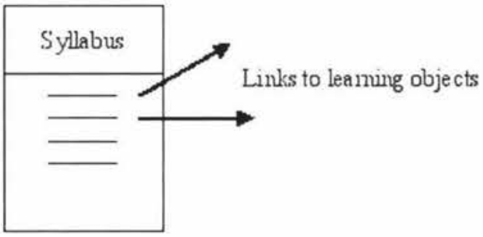


Figure 3.3: Schematic representation of a syllabus structure (Boyle, 2003)

In summary, the approach of sequencing Learning Objects in the multimedia Learning Object project is: (1) based on Syllabus, which is a pre-defined course structure, (2) the compound object structure allows a learning resource to launch another learning resource, and (3) it provides the Learning Objects sequencing or integration at a concept level.

3.1.3 Summary

Although the two above-mentioned learning systems are different in terms of the scope and functionality, the sequencing approaches applied in these systems have shown some significant similarities, which can also be found in other learning systems.

- Instructor manually predefines the course structure either in the content package (SCORM) or in the syllabus (MLO).
- The Dynamic course largely relies on either the effort of the instructor to rearrange learning resources (MLO, SCORM) or the possibilities of LMS (SCORM).

The common weakness of the approaches is that the Sequence Mechanism, which is the key to the presentation and delivery of the learning content, is pre-described in the form of a content package or a syllabus. The Learning Object contents can only be reused after the instructor or the learning system designer has spent time reviewing, evaluating and selecting it. The selected Learning Object is then put into a new content package or a new syllabus. The original contents sequence algorithm or pattern is no longer applicable, this work must be redone. Furthermore, without the instructor's review, evaluation, selection and rearrangement, the pedagogically relevant reuse of a Learning Object is impossible.

3.2 Necessity of the New Framework

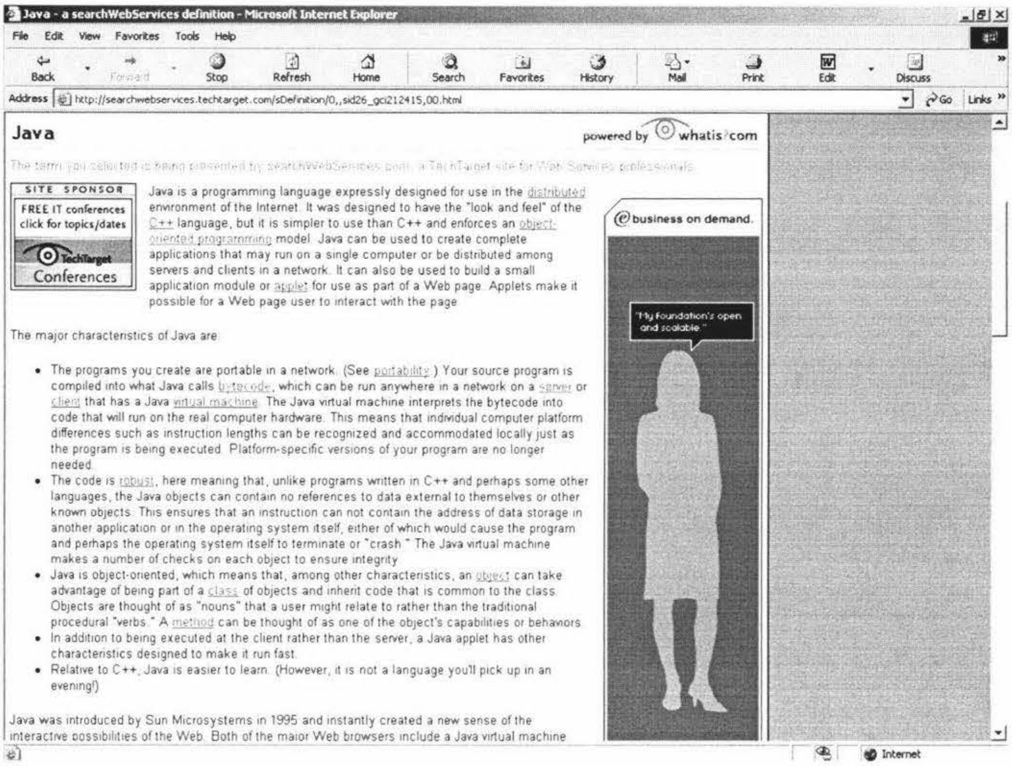
This section considers a scenario or learning circumstance in which a system-automated integration of Learning Objects is necessary. The goals and values of the integration process will also be addressed.

3.2.1 Scenario

In a self-paced, self-learning environment, where there is no instructor to manually arrange learning contents, all the Learning Objects are stored and ready for use in a Learning Objects repository from which the learner can search, locate and retrieve appropriate objects.

In a self-learning environment, in most cases the learner will encounter some other concepts or topics during the study, which are prerequisites of or related to the current learning concept. For example, in the case of searching "JAVA" in the whatis.com® web site http://whatis.techtarget.com/, whatis.com® is a "knowledge exploration and self-education tool about information technology" (whatis.com, 2003), as shown in Figure 3.4, "JAVA" related and supplementary concepts, such as "object-oriented programming", "distributed", "applet", "class", "object", "virtual machine", "robust" and "portability" are highlighted and presented with hyperlinks. This approach provides the learner options for further study and enables the learner to understand the concept in a context within a particular domain rather than as an isolated term.

However, if these concepts are not highlighted and accessible, the learner has to figure out which are important and related concepts in order to understand the "JAVA" concept. Similarly, in a self-learning environment, without the integration of decontextualized Learning Objects, the learner interacts with isolated concepts. To learn a related or supplementary concept, the learner has to search the repository again to get another decontextualized Learning Object. It is obviously an inefficient approach. In this self-paced, self-learning environment, what can Learning Objects do to provide the learner with a pedagogically rich learning experience and a higher-level of learning?



Java - a searchWebServices definition - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http://searchwebservices.techtarget.com/sDefinition/0,,sid26_gci212415,00.html
Go Links
powered by whatis?com
SITE SPONSOR
FREE IT conferences click for topics/dates
TechTarget Conferences
business on demand.
"My foundation's open and scalable."
Internet

Java

The term you selected is being presented by searchWebServices.com, a TechTarget site for Web Services professionals.

Java is a programming language expressly designed for use in the distributed environment of the Internet. It was designed to have the "look and feel" of the C++ language, but it is simpler to use than C++ and enforces an object-oriented programming model. Java can be used to create complete applications that may run on a single computer or be distributed among servers and clients in a network. It can also be used to build a small application module or applet for use as part of a Web page. Applets make it possible for a Web page user to interact with the page.

The major characteristics of Java are:

- The programs you create are portable in a network. (See portability.) Your source program is compiled into what Java calls bytecode, which can be run anywhere in a network on a server or client that has a Java virtual machine. The Java virtual machine interprets the bytecode into code that will run on the real computer hardware. This means that individual computer platform differences such as instruction lengths can be recognized and accommodated locally just as the program is being executed. Platform-specific versions of your program are no longer needed.
- The code is robust, here meaning that, unlike programs written in C++ and perhaps some other languages, the Java objects can contain no references to data external to themselves or other known objects. This ensures that an instruction can not contain the address of data storage in another application or in the operating system itself, either of which would cause the program and perhaps the operating system itself to terminate or "crash." The Java virtual machine makes a number of checks on each object to ensure integrity.
- Java is object-oriented, which means that, among other characteristics, an object can take advantage of being part of a class of objects and inherit code that is common to the class. Objects are thought of as "nouns" that a user might relate to rather than the traditional procedural "verbs." A method can be thought of as one of the object's capabilities or behaviors.
- In addition to being executed at the client rather than the server, a Java applet has other characteristics designed to make it run fast.
- Relative to C++, Java is easier to learn. (However, it is not a language you'll pick up in an evening!)

Java was introduced by Sun Microsystems in 1995 and instantly created a new sense of the interactive possibilities of the Web. Both of the major Web browsers include a Java virtual machine

Figure 3.4. JAVA and related concepts at the whatis.com® web site

3.2.2 Goals

A vision for the integration of Learning Objects is that the Learning Objects themselves should play more important roles in the automating process, especially in the above-mentioned self-learning circumstance, where no instructor will preview and pre-sequence the learning resources. When a group of instructors or course designers design a course for a particular domain or discipline, the instructors or designers are expected to map out a set of concepts and set up the relationships among them. When a learner provides information to the search engine to retrieve a Learning Object, as illustrated in

Figure 3.5, the Learning Object will detect other related Learning Objects, then present optional links to references for further study. It also can provide multiple presentations for the current learning concept, or utilise functionalities provided by other Learning Objects to solve a particular problem. This approach eventually provides the learner with a rich learning experience, which results in the augmentation of learner's knowledge.

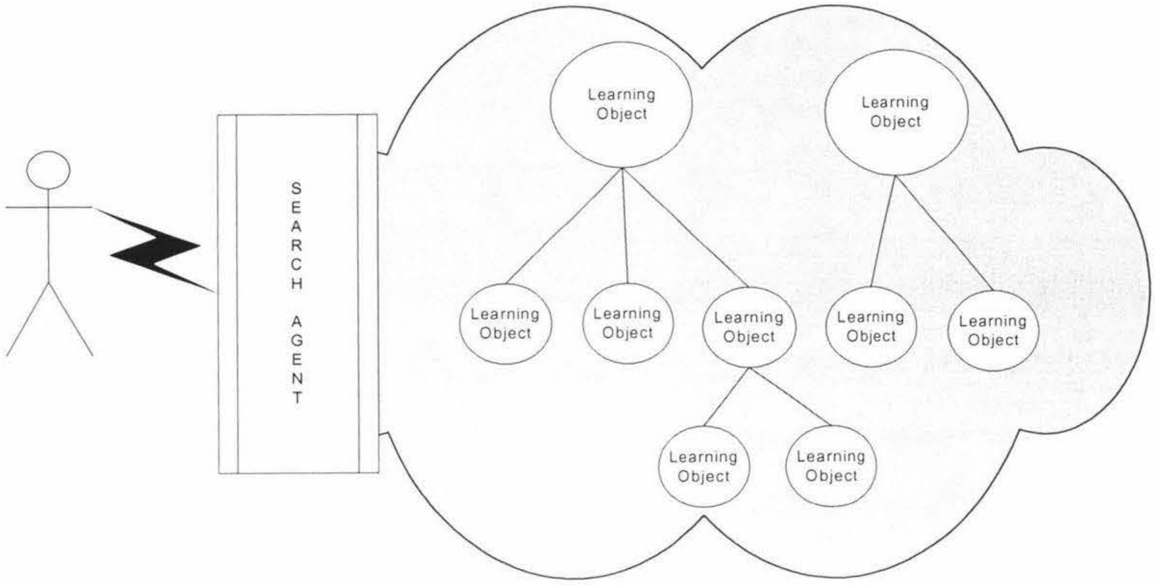


Figure 3.5. Integrated Learning Objects in the self-learning environment

As described above, Learning Objects have to play important roles and carry out intelligent tasks in the integration process. Therefore, learning contents in basic forms, such as images, sound and text cannot be used for the system-automated integration process. These Learning Objects should be an intelligent entity, which has embedded functionalities to carry out the intelligent tasks above-mentioned. An Intelligent Tutoring

Tool type Learning Object, as described in Chapter 2, is an intelligent entity, in which the required features for the integration can be extended and implemented.

In order to realize these goals, the following proposed integration framework requires an Intelligent Tutoring Tool type-Learning Object and is not applicable to other types of Learning Objects; hence, it is named Intelligent Tutoring Tools Integration Framework (ITTIF).

3.2.3 Values

The values of the integration of ITTs are obvious. These are:

- ❖ Extend the scope of a single ITT
- ❖ Contextualize independent ITT
- ❖ Utilise the functions provided by other ITTs to solve a particular problem
- ❖ Provide related learning
- ❖ Enable a higher level of learning

With the vision of integrated Intelligent Tutoring Tools, developing a structure that will enable them to be used becomes important. The following section will have a detailed look at a framework, which will answer the question "how to integrate the Intelligent Tutoring Tools?"

3.3 Characteristics of the ITT Integration Framework

Kinshuk & Patel (1997) have proposed a generic software structure for ITT design and development, including the instructional design, Interface design, ITT Integration and so forth. They also have suggested a structure to integrate Intelligent

Tutoring Applets (ITA). The ITA is an implementation of ITT on the Internet. It shows that the linkage of multiple Intelligent Tutoring Applets (ITAs) created by different teachers enables the creation of Intelligent Tutoring Systems (ITS). The structure is illustrated in Figure 3.6.

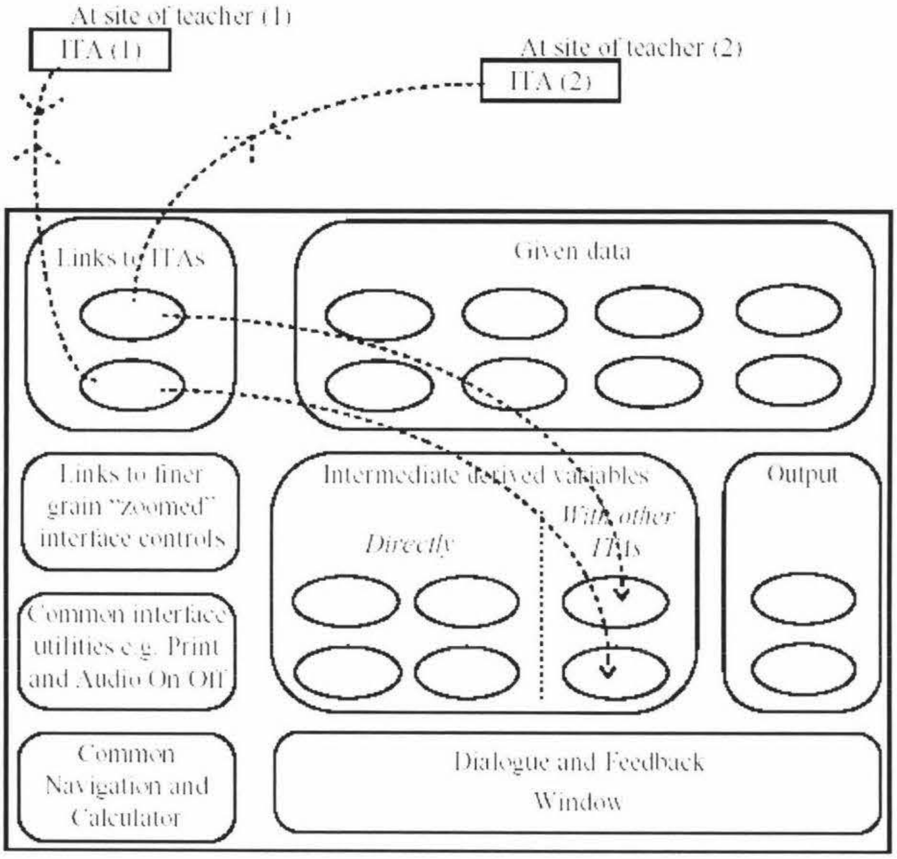

At site of teacher (1)
ITA (1)
At site of teacher (2)
ITA (2)
Links to ITAs
Given data
Links to finer grain "zoomed" interface controls
Intermediate derived variables
Directly
With other ITAs
Output
Common interface utilities e.g. Print and Audio On Off
Common Navigation and Calculator
Dialogue and Feedback Window

Figure 3.6. Intelligent tutoring system on Internet and its linkages with various ITAs (Kinshuk & Patel, 1997)

The fundamental concept and idea is also applicable to the Intelligent Tutoring Tools Integration Framework. However, ITTIF focuses on and formalizes the integration of ITTs. It contains four components, as illustrated in Figure 3.7.

- Firstly, a shareable explicit specification of conceptualization, ontology, which is the basis for integration and collaboration.
- Secondly, in order to identify structured sequence or contextual relationships between ITTs, a formalized design and sequence theory needs to be adapted.
- Thirdly, standardised metadata should be available to describe the context relationship between ITTs.
- Finally, an integration mechanism should be proposed, such as relationship detector, relationship activator.

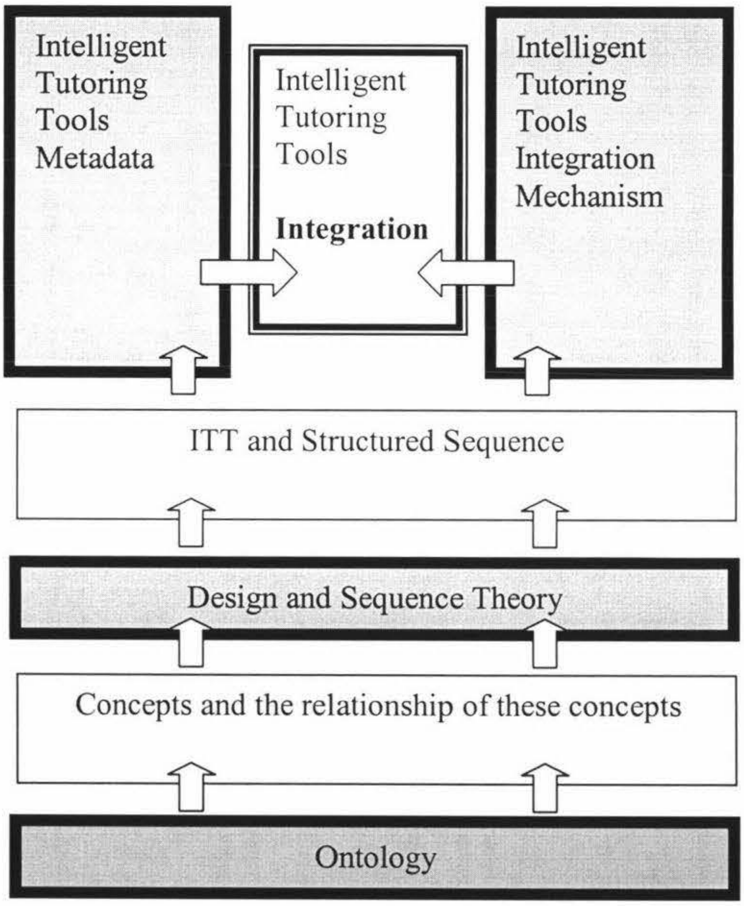

Intelligent Tutoring Tools Metadata
Intelligent Tutoring Tools
Integration
Intelligent Tutoring Tools Integration Mechanism
ITT and Structured Sequence
Design and Sequence Theory
Concepts and the relationship of these concepts
Ontology

Figure 3.7. ITT integration framework

A detailed discussion about the four components is presented below.

3.3.1 Ontology

Why is ontology necessary in the integration of ITTs?

The term "Ontology" is from philosophy, where it is a systematic account of existence. When it is borrowed and used in Artificial Intelligence systems, it is defined as "an explicit specification of a conceptualization" (Gruber, 1993). In the same work, Gruber stated that the conceptualization was "an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly". Furthermore, the conceptualization consists of the objects, concepts, and other entities, as well as the relationships that hold among them (Gruber, 1993). In a word, the ontology is used to share the understanding of a domain of interest and it reflects a set of concepts, and the describable relationships between them in a representational vocabulary.

Thuraisingham (2002, p116) emphasised that ontology is needed "whenever two or more people have to work together". He also gave a general example of the usage of ontology, "different groups [were] collaborating on a design project. They could define ontologies so that they all spoke the same language. If ontologies were previously defined by other design groups, they could reuse these ontologies to save time" (2002, p116). A real world example can be found in the traditional Electric Data Interchange (EDI) application, a set of specifications for goods, such as price and product model number have to be standardized between trading partners before exchanging any business data. In the case of Intelligent Tutoring Objects design, different instructors are collaborating on the learning resource design for a particular domain, and so a common language, a shareable understanding regarding the domain of interest has to be defined.

Not only does a well-defined ontology in a domain of interest benefit the collaboration of instructor designers, but also it gives the possibility for the integration of independent Intelligent Tutoring Tools since the independent ITTs share a common set of conceptualizations.

3.3.2 Instructional Design and Sequence Theory

Having a set of concepts and the describable relationships among them enables a computer system to present the learner an appropriate learning concept, as well as the related concepts based on their search criteria. However, this simple "linking" mechanism is insufficient to provide a pedagogically rich learning experience; meanwhile, it may cause another information overload problem. In order to identify structured sequences and provide contextual relationships between ITTs, a formal instructional design and sequence theory needs to be adapted.

Wiley (2000) proposed a Learning Object Design And Sequencing Theory (LODAS), after reviewing, synthesizing, and combining four existing instructional design theories, Elaboration Theory, Work Model Synthesis, Domain Theory, and the Four-Component Instructional design model with his own work. Wiley (2000) stated that LODAS had clear goals to support the design, sequence and reuse of Learning Objects and it functioned best in a learning environment where instructors empower learners in their own learning towards a complex cognitive skill, which is similar with and applicable to the self-learning environment. LODAS is selected and applied in the ITTIF.

LODAS consists of the following six processes (Wiley, 2000). These are:

- Preliminary Activities

  Determine appropriateness

- Content analysis and synthesis

  Transform an undifferentiated content domain into specifications of the content scope and sequencing of Learning Objects

- Practice and information presentation design

  Specify the specific problems and instruction which will be instantiated in Learning Objects

- Learning object selection or design

  Specify types of Learning Objects and provide guidance for the design of each Learning Object type

- Learning object sequencing

  Provide instructional sequencing specifications for the Learning Objects

- Loop back for quality improvement

  Implementation, evaluation and revision

Since the ITT Integration framework focuses on the integration of ITTs, the highly related processes of content analysis and synthesis and Learning Object sequencing, are briefly reviewed here.

- Content analysis and synthesis

This process consists of a group of methods, including the "principled skill decomposition" method which breaks a complex cognitive skill down into its constituent skills, and the "Synthesize work model" method which then recombines the decomposed constituent skills into work models. A work model is defined in LODAS as "activities that real people perform in the real world" (Wiley, 2000, p.60). The two methods are illustrated in Figure 3.8.

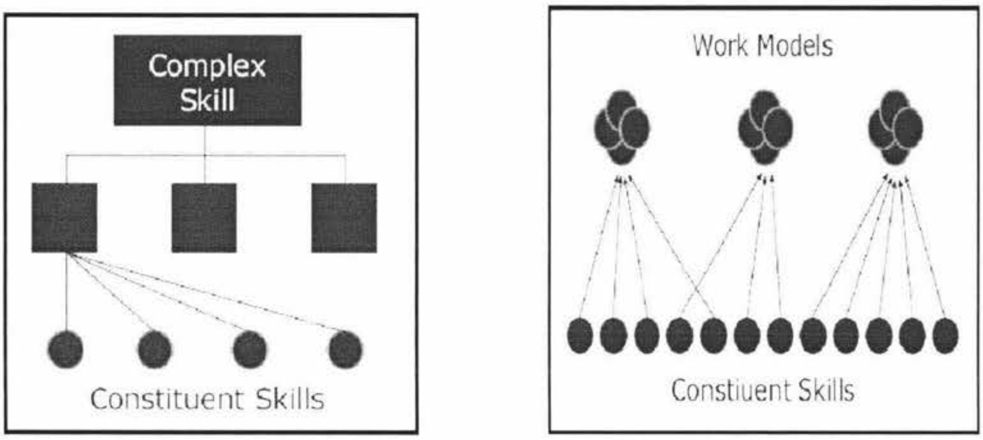


Figure 3.8. Principled skill decomposition. (Wiley, 2000, p.60) and Work model synthesis. (Wiley, 2000, p61)

Other methods in this process include: "identifying the dimensionality of the domain", "placing work models on scales", "synthesizing integrated work models" and "exposing a domain map to expert review". Finally, a well-defined domain map, in which work models are placed on the identified dimensional scale of expertise in order of increasing complexity, is presented as a blueprint for the instructional design (see Figure 3.9).

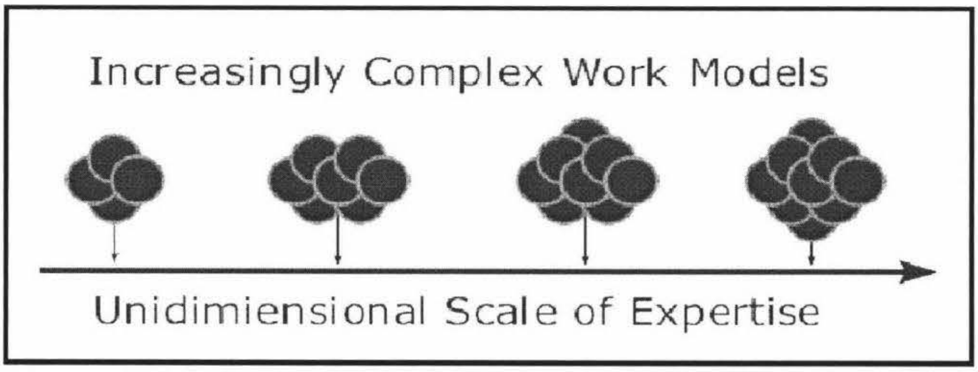

Increasingly Complex Work Models
Unidimiensional Scale of Expertise

Figure 3.9. Domain map with a unidimensional scale of expertise (Wiley, 2000, p.63)

- Learning object sequencing

In LODAS, the Learning Object sequencing occurs on three levels: work models, case types, and specific problems. Case types are "practice specifications based on simple to complex versions of the work model", such as instructional problems and worked examples within the work model. Specific problems are "instantiations of the case types" (p.71). Learning Objects representing work models and case types are sequenced in a simple to complex ordering, while those representing specific problems within a case type are on a single level in a random order.

3.3.3 Metadata

The Intelligent Tutoring Tools and the relationships and sequencing information between them identified above should be defined in a standardised metadata, which is external to the Intelligent Tutoring Tools.

LOM (2002) based on Dublin Core defined a "relation" category, which defines the following relationships between Learning Objects as shown in Table 3.1. This category is applicable to the ITTIF and will be used to describe the relationships between Intelligent Tutoring Tools.

Table 3.1.

Relation Category in Learning Object Metadata (LOM, 2002)

Element Name	Explanation	Relationship
Relation.Kind	Nature of the relationship between this Learning Object and the target Learning Object	ispartof: is part of
		haspart: has part
		isversionof: is version of
		hasversion: has version
		isformatof: is format of
		hasformat: has format
		references: references
		isreferencedby: is referenced by
		isbasedon: is based on
		isbasisfor: is basis for
		requires: requires
		isrequiredby: is required by

3.3.4 Integration Mechanism

Finally, an integration mechanism should be implemented, which enables the Intelligent Tutoring Tools to access the metadata and detect and activate the relationship with other Intelligent Tutoring Tools.

- Relationship Detector

  Retrieve relationship information defined in the metadata. Based on this information, locate and detect the existence of the related Intelligent Tutoring Tools.

- Relationship Activator

  If a related Intelligent Tutoring Tools exists, activate the linkage to the related ITTs.

3.4 Summary

In the Intelligent Tutoring Tools Integration Framework (ITTIF), the presentation sequence and relationships between Intelligent Tutoring Tools are expressed in the explicit specification of conceptualization and refined by the adaptation of the LODAS theory. Secondly, they are described in the metadata which is external to the learning content. Finally, the relationships between these ITTs are detected by the relationship detector, and the links are only activated in the case that related ITTs exist. The proposed framework is expected to provide an Intelligent Tutoring Tool level integration that will extend the scope of an independent ITT and guide the learner from a simple work model to a complex work model, resulting in a higher-level of learning.

To examine the proposed framework, an Intelligent Tutoring Tools integration prototype system is implemented and described in detail in the next chapter.

Chapter 4

Toward the Prototype of Intelligent Tutoring Tools Integration System

Chapter 3 has overviewed some Learning Objects systems focusing on different approaches to integrate independent Learning Objects, and proposed an Intelligent Tutoring Tools Integration Framework (ITTIF) to provide the learner with a pedagogically rich learning experience in a self-learning environment. In order to examine the framework, an Intelligent Tutoring Tools integration system (prototype) is implemented. In this chapter, a brief introduction to the prototype development method is presented, followed by a detailed description of the prototype implementation, including initial analysis, objective definition, specification and construction.

4.1 Prototype Development

The traditional system development life cycle, which is also known as a waterfall life cycle, consists of individual phases. These are system planning, system analysis, design, implementation, testing, installation and maintenance. Each phase has explicitly defined outputs, which in turn become the input of a subsequent phase. For example, at the end of the analysis phase, a set of completed specifications, such as the users' requirements, system function and acceptance test, are produced and are requisites for the next design phase.

However, in reality, the waterfall approach has a significant difficulty, because the users' requirements are very hard to finalize at an early stage of the system development. The changes to users' requirements can happen in the design, implementation and other subsequent phases when the user gets further involved in the system development. The prototype approach is expected to involve the user at the early stage of the system development to overcome the potential misunderstandings between developers and users to help the user to clearly define the system requirements; therefore, a high quality system development can be achieved.

A prototype is "a smaller-scale, representative or working model of the users' requirements or a proposed design for an information system" (Whitten, Bentley & Dittman, 2001, p98). It is not intended as a final system, as Bennett, Mcrobb & Farmer (2002, p51) stated "a prototype system is differentiated from the final production system by some initial incompleteness and perhaps by a less resilient construction". The main objective of the prototype system development is building a working model quickly and commonly involves the adaptation of rapid development tools, to explore and examine some core aspects of the proposed final system. The core aspects can be users' requirements, human-computer interface design, efficacy of a particular development language or a database management system, and so forth.

As a working model of the final system, a prototype system development has its own life cycle (see Figure 4.1). The main stages required to implement a prototype are described as follows (Bennett, et al., 2002, p52):

- ❖ Perform an initial analysis

  An initial analysis is conducted to outline the system requirements.

- Define prototype objectives

  Prototypes may be constructed for various purposes. For each prototype, it has a clearly defined objective.

- Specify prototype

  The scope of the prototype should be clearly specified. Within the scope, analysis and design methods are applied.

- Construct prototype

  A working prototype is constructed with some development tools and delivered for evaluation.

- Evaluate prototype and recommend changes

  Users provide feedback for enhancement.

These main stages, except the final evaluation, are described in detail in the following sections in this chapter. The final evaluation is covered in Chapter 5.

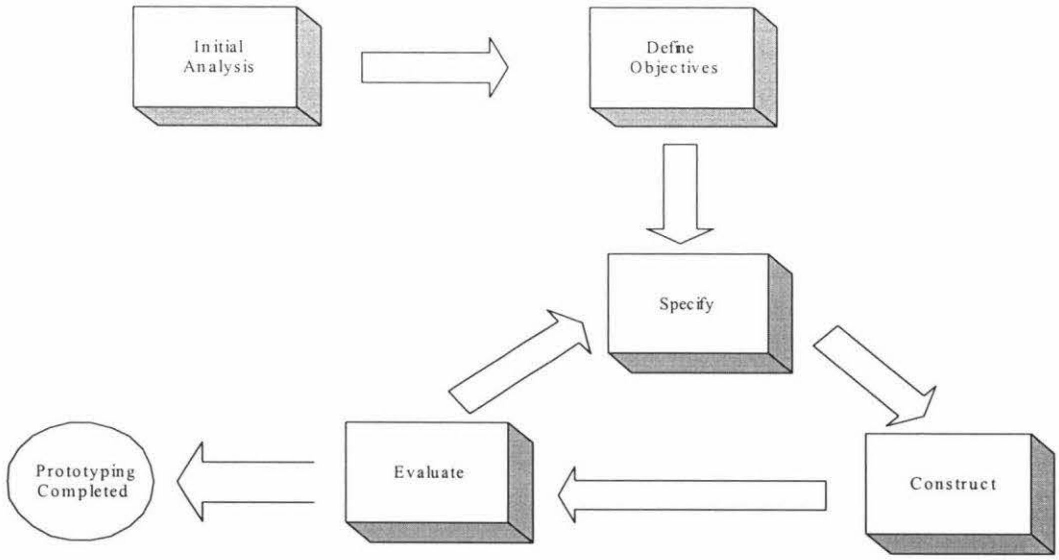


Figure 4.1. A prototype life cycle (Bennett, et al., 2002, p52)

4.2 Initial Analysis

Object-oriented analysis consists of four general activities and they are as follows: (Whitten, et al., 2001, p656)

1. Modelling the functions of the system.
2. Finding and identifying the business objects
3. Organizing the objects and identifying their relationships
4. Modelling the behaviour of the objects

During the initial analysis phase, the functional aspects of the system are modelled and presented in the use case model diagram. After setting up the project objective, other analysis activities are covered in the subsequent prototype specification stage.

Chapter 3 discussed a self-paced, self-learning environment, for instance, the Learning Objects repository. Three types of users interact with the learning system. They are: (1) instructors or course designers who collaboratively design and create learning contents for a particular domain, which are then put into the system, (2) system administrators who manage and maintain these contents, and (3) learners who give information to the search engine to find and use the required Learning Objects. Therefore, a list of actors and use cases for the Intelligent Tutoring Tools integration system are identified and shown in Table 4.1.

Table 4.1.

Actors And Use Cases In The Intelligent Tutoring Tools Integration System

Actor	Use Case Name	Use Case Description
Learners	Search Learning Contents	Learners search the learning contents via search engine
Learners	Use Learning Contents	Learners use the selected learning contents
Instructors	Generate Learning Contents	Instructors use authoring tools or templates to generate learning contents
Instructors	Publish Learning Contents	Instructors upload and put learning contents into the Learning Objects repository
Administrators	Manage Learning Contents	Administrators manage and maintain learning contents

Once the use cases and actors have been identified, the use case model diagram can be used to graphically depict the system scope and boundary. The use case model diagram for the use cases listed in Table 4.1 is presented in Figure 4.2.

4.3 Prototype Objective

This prototype demonstrates the integration of independent Intelligent Tutoring Tools, which can be developed by different instructors and learning system designers. The integration of independent Intelligent Tutoring Tools, as described in Chapter 3, is one of the core functional requirements of the learning system.

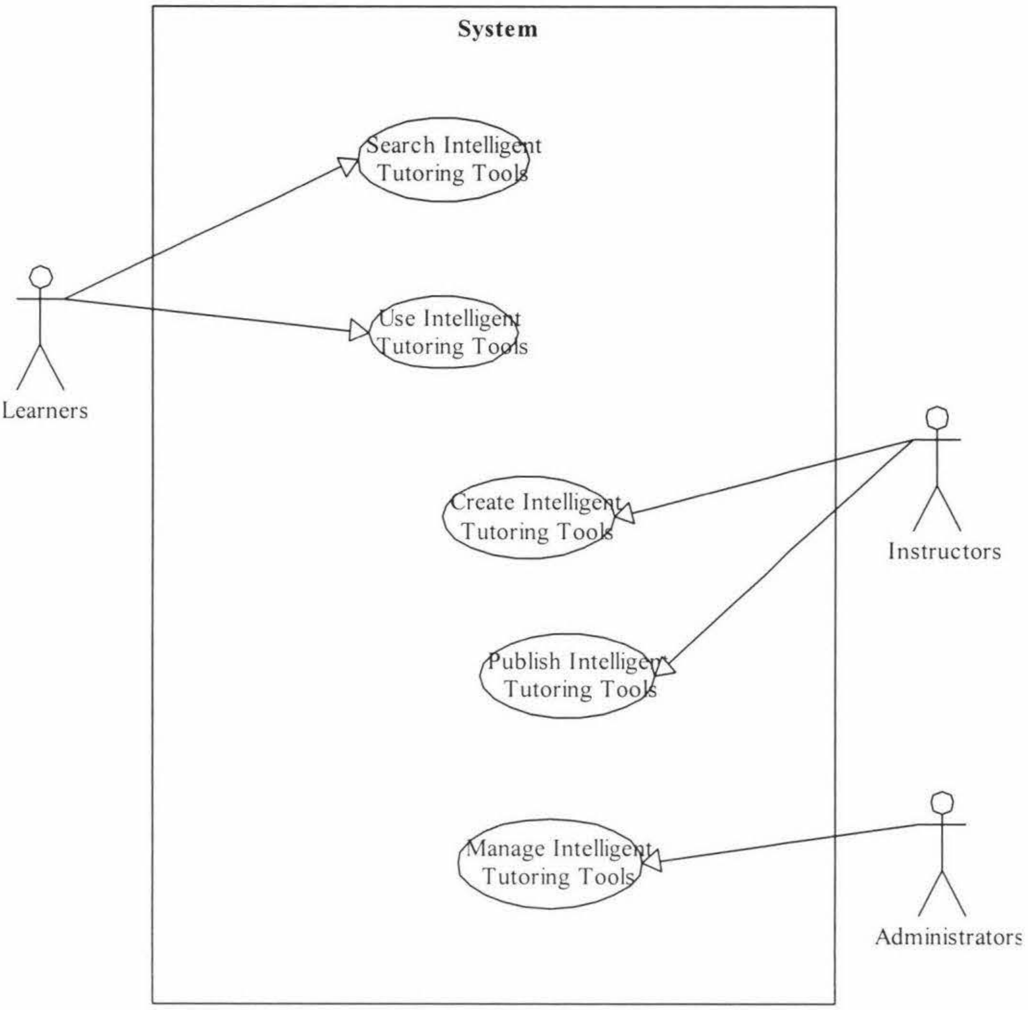

System
Search Intelligent Tutoring Tools
Use Intelligent Tutoring Tools
Learners
Create Intelligent Tutoring Tools
Instructors
Publish Intelligent Tutoring Tools
Manage Intelligent Tutoring Tools
Administrators

Figure 4.2. Intelligent tutoring tools integration system use case diagram

The Intelligent Tutoring Tools Integration Framework consists of four components: ontology, Learning Object design and sequencing theory, metadata and integration mechanism. These are applied in the prototype development in the following ways:

- The framework enables finding and identifying the Learning Objects.

- The framework enables organizing the Learning Objects and identifying their relationships in a pedagogical way.
- The framework enables automating the integration of the learning objects.

4.4 Specify a Prototype

The scope of the prototype should be clearly specified at this stage.

The textbook, "Interactive Accounting: The Byzantium Workbook" (Wilkinson-Riddle & Patel, 1997), was developed to introduce a complete course in financial and management accounting techniques, for example, balance sheet and standard costing. From the course, a module introducing the capital investment appraisal technique in management accounting was selected and implemented in the prototype.

Within this clearly specified scope, a further detailed system analysis and design is described below.

4.4.1 System Analysis

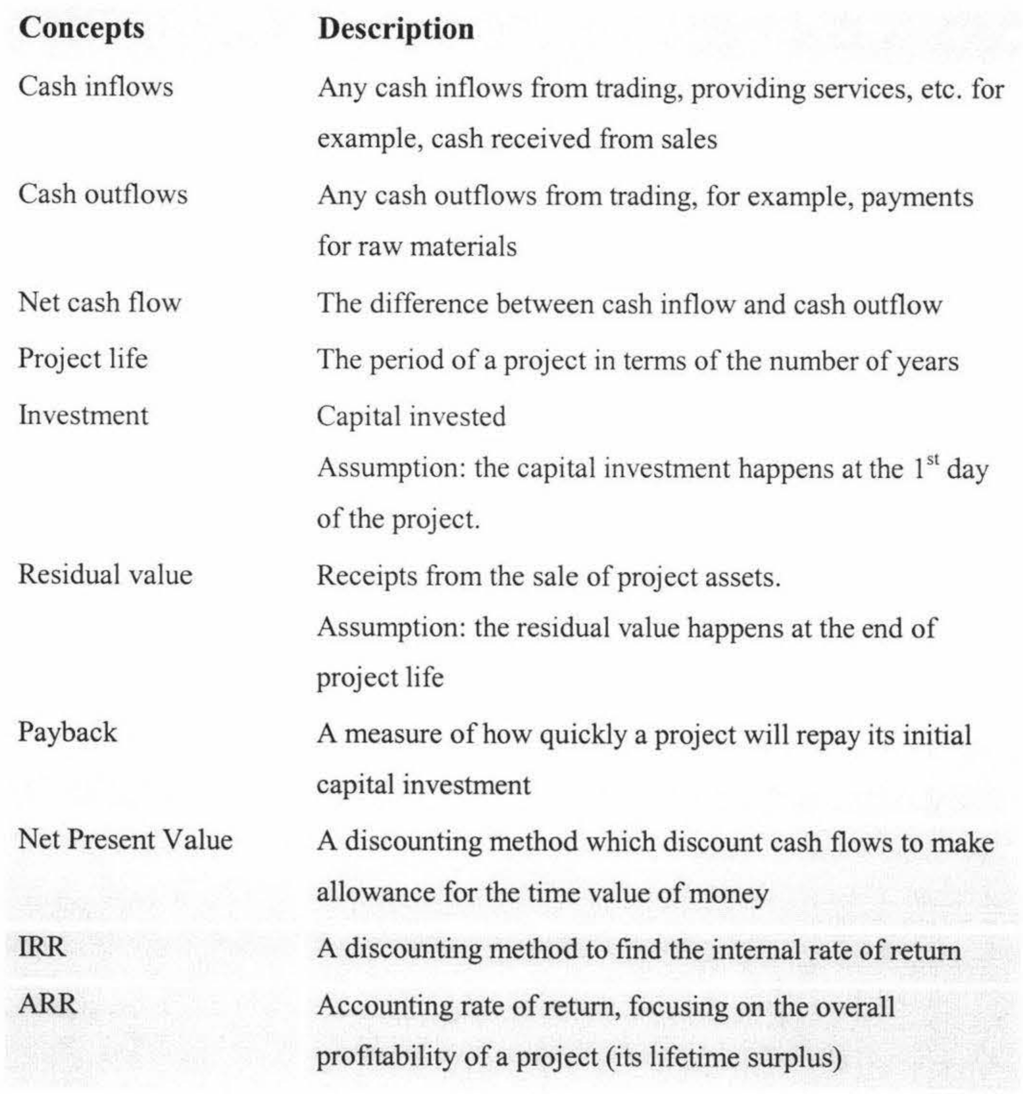
Larman (2002) stated that object-oriented analysis emphasized "finding and describing the objects or concepts in the problem domain". In order to find Learning Objects and identify their relationships, first, a common explicit conceptualization of the domain needs to be set up.

4.4.1.1 Ontology

Within this specified capital investment appraisal in the management accounting domain, concepts and inter-relationships need to be identified as a shareable common set of conceptualization. A subset of the concepts is summarized in Table 4.2,

Table 4.2.

A Subset Of Concepts In Capital Investment Appraisal Domain



Concepts	Description
Cash inflows	Any cash inflows from trading, providing services, etc. for example, cash received from sales
Cash outflows	Any cash outflows from trading, for example, payments for raw materials
Net cash flow	The difference between cash inflow and cash outflow
Project life	The period of a project in terms of the number of years
Investment	Capital invested Assumption: the capital investment happens at the 1st day of the project.
Residual value	Receipts from the sale of project assets. Assumption: the residual value happens at the end of project life
Payback	A measure of how quickly a project will repay its initial capital investment
Net Present Value	A discounting method which discount cash flows to make allowance for the time value of money
IRR	A discounting method to find the internal rate of return
ARR	Accounting rate of return, focusing on the overall profitability of a project (its lifetime surplus)

Besides the list of concepts in the domain, an inter-relationships network of the concepts in the domain can be identified; Figure 4.3 demonstrates a partial inter-relationships network, which is based upon the following equations as well as the concepts defined in Table 4.2:

$$NCF_i = CIF_i - COF_i$$

Where NCF stands for Net Cash Flow, CIF stands for Cash inflows, COF stands for Cash outflows and 'i' is the number of years.

$$CNF = \sum_{i=0}^{n} NCF_i$$

Where CNF stands for Cumulated Net Flow, 'n' is project life in terms of number of years.

$$PV_i = FV_i * \frac{1}{(1+IRate)^i}$$

Where PV stands for Present Value, FV stands for Future Value, IRate stands for Interest Rate.

$$NPV = \sum_{i=1}^{n} PV_i$$

Where NPV stands for Net Present Value.

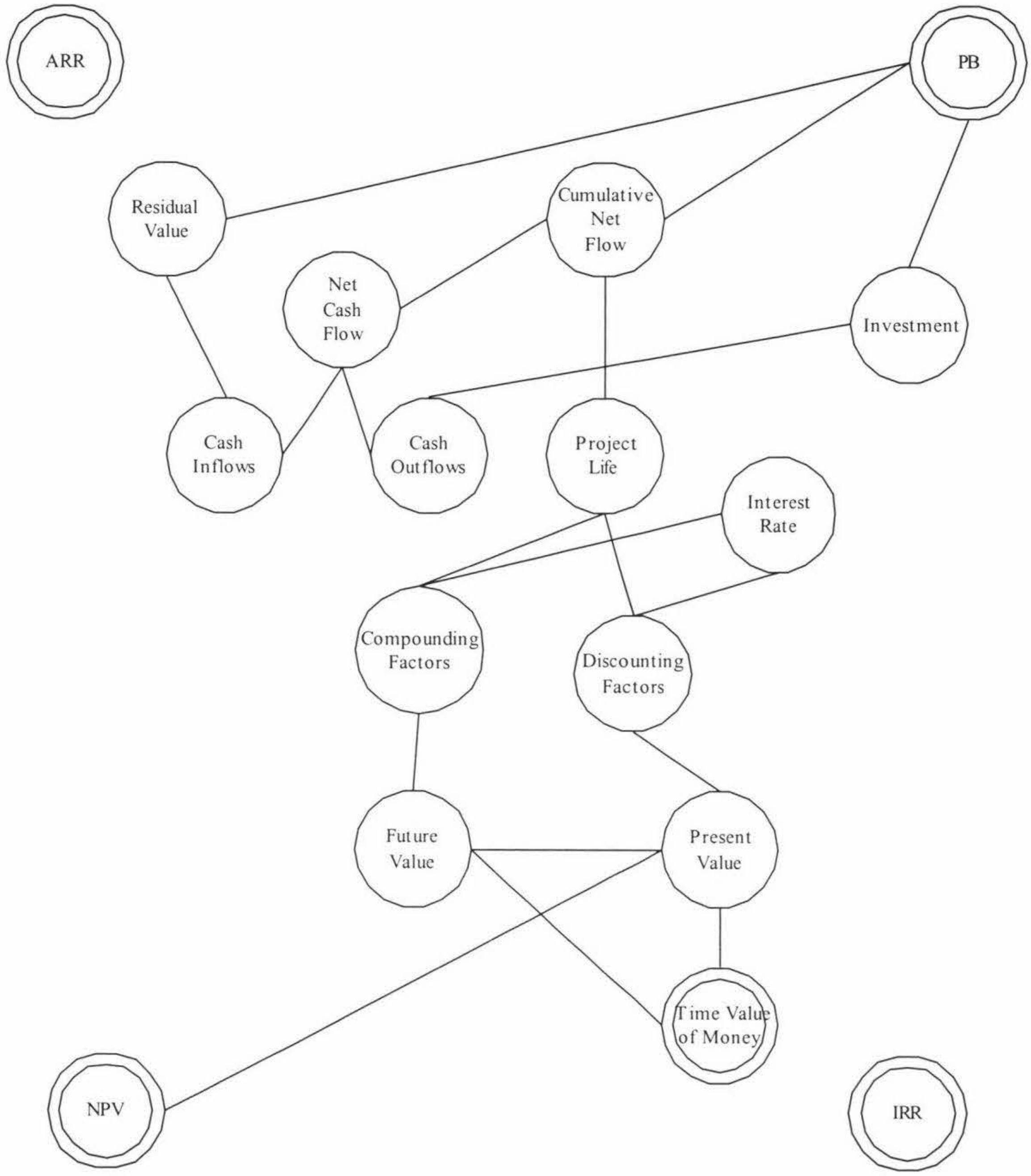


Figure 4.3. A subset of the network of inter-relationships for capital investment appraisal

In this section, a shareable ontology is defined to describe the concepts and the interrelationships between them in the capital investment appraisal domain. The next section shows that the design and sequence techniques from LODAS are applied to identify and organize Learning Objects.

4.4.1.2 Learning Objects Design and Sequence

The principled skill decomposition, work models synthesize and domain dimensionality identification proposed by LODAS are applied and described in this section.

1. Principled skill decomposition

The complex cognitive skill, capital investment appraisal, to be taught and implemented in the prototype system is broken down into its constituent parts as below, see Figure 4.4.

- ❖ Identify project information, for example, the investment, residual value, project life, cash inflow and cash outflows, etc.
- ❖ Understand the capital investment appraisal concepts and methods, for example, ARR, PB, NPV, and IRR, etc.
- ❖ Understand the difference and relationships between these methods, for example, Profitability focused method, Cash Flow focused method and Cash flow focused Discount methods, etc.
- ❖ Identify the circumstance of applying these methods

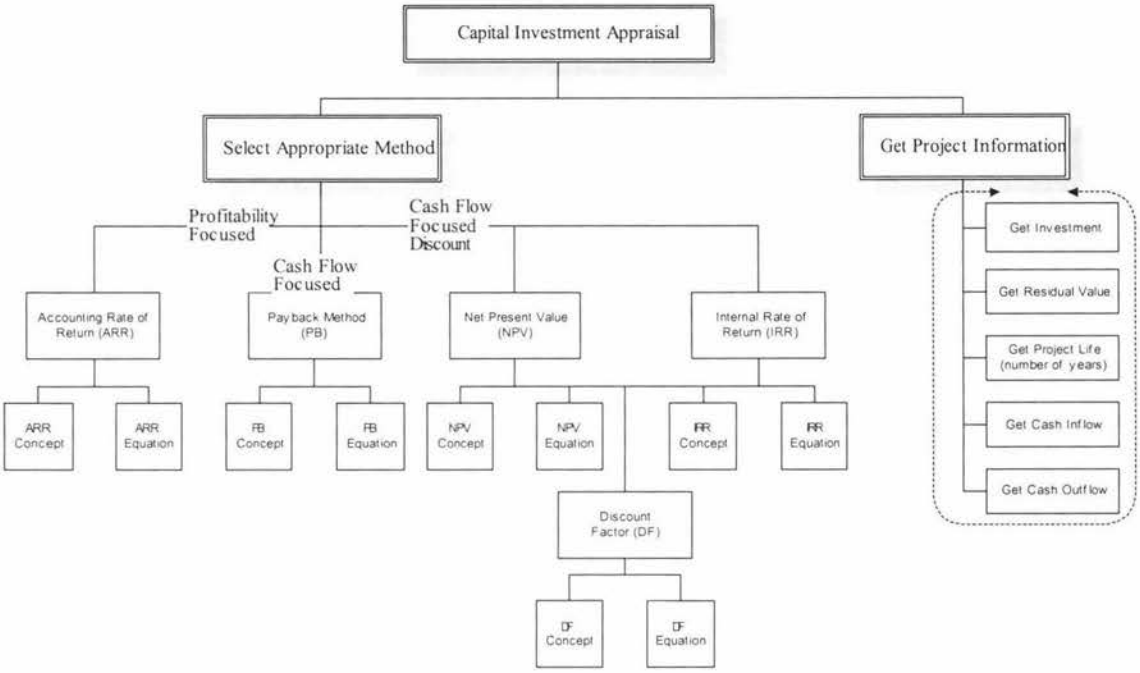


Figure 4.4. Principled skill decomposition of capital investment appraisal

"The vertical relationship, which is indicated from bottom-to-top between child skills on a certain level and their parent skill one level higher, signifies that constituent skills lower in the hierarchy enable or are prerequisite to the learning and performance of skills higher in the hierarchy." (van Merriënboer, Clark, and de Croock, 2002) For example, you must be able to understand the Discount Factor concept and know how to calculate it in order to use the Net Present Value or Internal Rate of Return methods to appraise capital investment.

2. Work models synthesis

The constituent skills are then recombined into activities that people perform in the real world as below. Figure 4.5 shows "Apply NPV method" and "calculate Discount Factor".

- Apply Accounting Rate of Return method
- Apply Payback method
- Apply Net Present Value method
- Apply Internal Rate of Return method
- Calculate Discount Factor

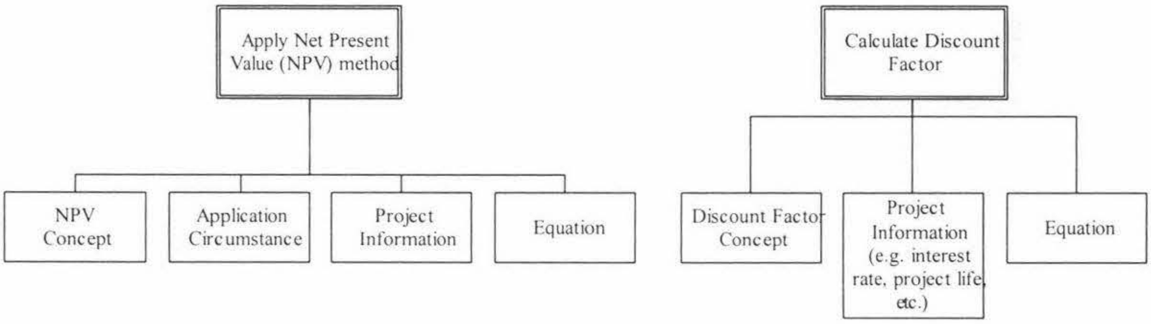

Apply Net Present Value (NPV) method
NPV Concept
Application Circumstance
Project Information
Equation
Calculate Discount Factor
Discount Factor Concept
Project Information (e.g. interest rate, project life, etc.)
Equation

Figure 4.5. Synthesized working model for capital investment appraisal

3. Identify the dimensionality of the domain and place work models on a scale of increasing complexity.

In the capital investment appraisal domain, only one dimensionality, appraise capital investment, is identified. The determination about the position of each work model on the scale is made according to the difficulty or relevance of each work model (see Figure 4.6).

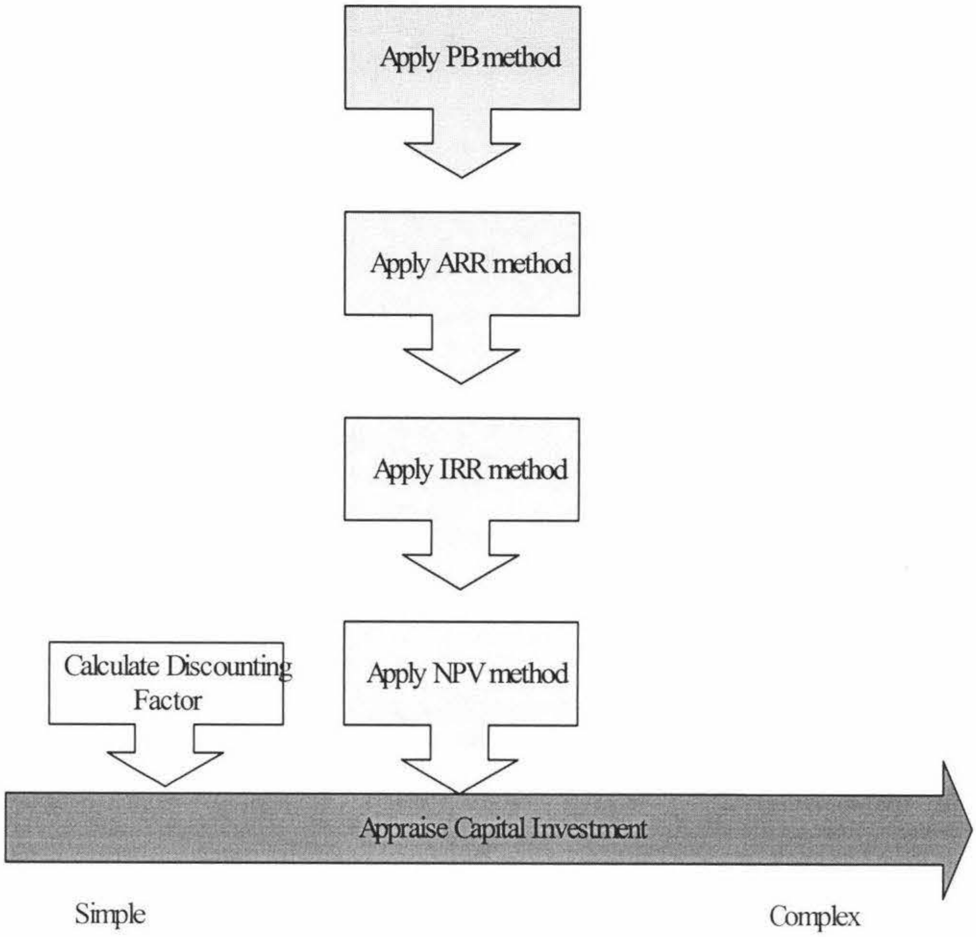


Figure 4.6. Work models with one-dimensional scale

During the system analysis, working models in the capital investment appraisal domain have been identified, synthesized and organized. The working models can be viewed and implemented as independent, self-contained Intelligent Tutoring Tools. The identified objects and the relationships between them are partially illustrated in a class diagram (see Figure 4.7), in which only the attributes of NPV and DF classes are defined.

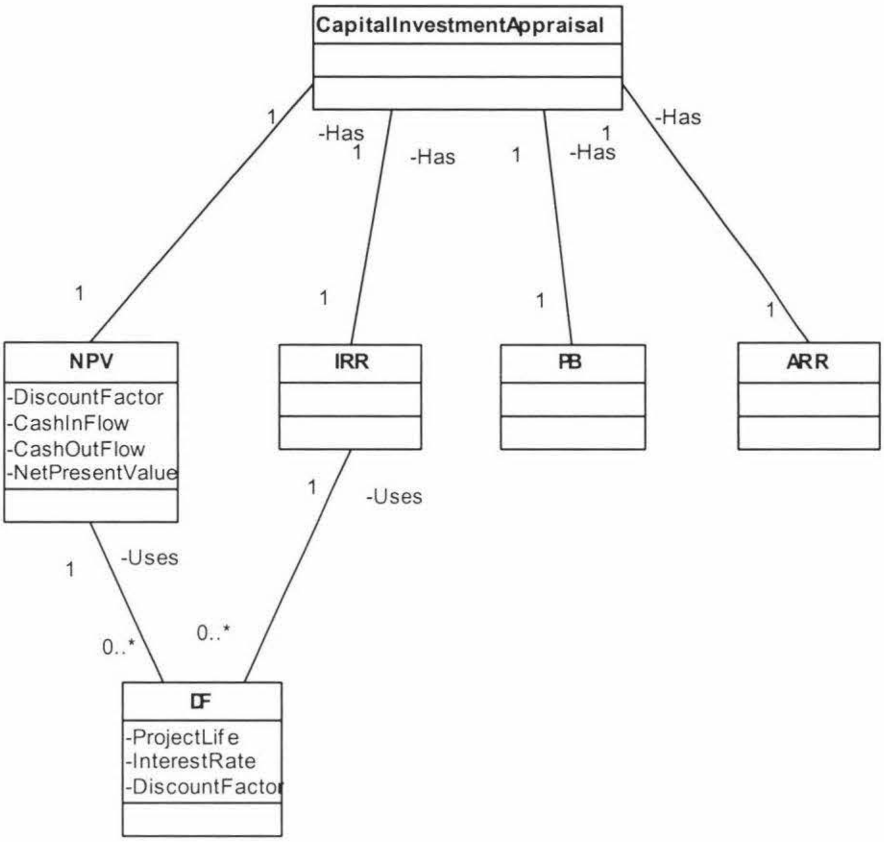


Figure 4.7. Partial class diagram for Intelligent Tutoring Tools integration system (capital investment appraisal)

4.4.2 System Design

Larman (2002) stated that object-oriented design emphasizes "defining software objects and how they collaborate to fulfil the requirements."

The system analysis concentrates on identifying the entity objects that represent actual data within the domain. The system design continues to (1) find the responsibilities of each object to refine these entity objects, and (2) introduce other objects, including interface objects that represent the system interfaces and control objects that represent the application logic. The structure of an object-based system, made up of entity, interface and control objects, is similar to the model-view-controller (MVC) mechanism (Whitten, et al., 2001, p676). These objects are finally implemented and collaborated to fulfil the system requirements.

First of all, a detailed use case correlating to the integrated "capital investment appraisal" learning is presented (see Table 4.3).

Table 4.3.

Use Case Correlating To The Integrated "Capital Investment Appraisal" Learning

Use Case Name:	"Capital investment appraisal" learning	
Actor(s):	Learner	
Description:	This use case describes the process of a learner using the learning resources about "capital investment appraisal"	
Typical Course of events	Actor Action	System Response
	The Capital Investment Appraisal window is currently displayed on the screen waiting for the learner to select one of the four methods. 1. Learner selects NPV method.	
		2. The Intelligent Tutoring Tools

		“NPV” will create a screen to present the NPV concept. Meanwhile, it will check the related concept. If there are some related concepts (Discount Factor & Capital Investment Appraisal), it will generate links (buttons), which guide the user to related concepts.
		3. The Intelligent Tutoring Tools “NPV” will create a screen to present the NPV calculation. Meanwhile, it will check the input set of NPV and the output set of other ITTs. If any input value of NPV can be provided by another ITT, it will activate the link, which allows the user to get the value from another ITT. For example, another “DF” ITT can provide the Discount Factor in the Net Present Value calculation.
	4. Learner selects “get” DF from other ITT	
		5. The system will generate a parameter screen based on the input set of ITT “DiscountFactor”, waiting for the learner entry
	6. Learner enters the “project life” and “interest rate”	
		7. The system will get the input and pass to ITT “DiscountFactor” for

<table>
<tr><td></td><td></td><td>calculation. The returned value will be set in the NPV calculation screen.</td></tr>
<tr><td></td><td>8. Learner enters other “cash inflows” and “cash outflows”, and clicks NPV button</td><td></td></tr>
<tr><td></td><td></td><td>9. The system will get the input and pass to ITT NPV for calculation. The returned value will be set in the screen.</td></tr>
<tr><td><b>Alternate Courses:</b></td><td colspan="2">4. Learner selects the related concept. The system will guide the learner to the related concept screen. (Discount Factor & Capital Investment Appraisal)</td></tr>
<tr><td><b>Precondition:</b></td><td colspan="2">All the related concepts, input set and output set of Intelligent Tutoring Tools have been identified and described.</td></tr>
<tr><td><b>Post condition:</b></td><td colspan="2">None at this time</td></tr>
<tr><td><b>Assumption:</b></td><td colspan="2">None at this time</td></tr>
</table>

As the precondition, all the related concepts, input and output set of Intelligent Tutoring Tools have been identified and described. As described in Chapter 3, the Learning Object Metadata has the “Relation” category, which can be used to describe the related concepts. However, the description of the input and output set of the Intelligent Tutoring Tool needs an extension of the LOM metadata. An example of the metadata with XML binding is presented in the prototype construction section.

Given this detailed use case, object behaviours and responsibilities can be identified. That means an object has to provide a service when requested, or collaborates with other objects to satisfy a request if required (Whitten, et al., 2001, p677). The prototype system, for simplicity, has combined the interface and control behaviour into a single kind of object. The responsibilities of main objects in the prototype system are summarized in Table 4.4.

Table 4.4.
A Subset Of The Main Objects And Their Responsibilities

Objects	**Types**	**Responsibility**

NPV	Model	setCashInflow():
		responsible for setting cash inflows
		setCashOutFlow():
		responsible for setting cash outflows
		setDiscountFactor():
		responsible for setting discount factor
		getCashInflow():
		responsible for getting cash inflows
		getCashOutFlow():
		responsible for getting cash outflows
		getDiscountFactor():
		responsible for getting discount factor
		compute():
		responsible for Net Present Value calculation

DF	Model	setProjectLife():
		responsible for setting project life in terms of number of years
		setInterestRate ():
		responsible for setting interest rate
		getProjectLife():
		responsible for getting project life in terms of number of years
		getInterestRate ():
		responsible for getting interest rate
		compute():
		responsible for Discount Factor calculation

Frame_CIA	View_control	main():

		responsible for starting the application.
Frame_NPV	View_control	getconcept(): responsible for the view of the NPV concept getrelatedconcept(): responsible for the links with related concepts getfunction(): responsible for the view and getting the input set of NPV calculation getrelatedfunction(): responsible for the output set of other ITTs activation(): responsible for activate the link in the case of related function available related_concept0(): responsible for linking related concepts (Note: In the prototype system, there are up to 5 related concepts implemented.) setdf(): Responsible for getting the user input and setting the calculated discount factor compute(): responsible for passing the user input to NPV object, and setting up the returned Net Present Value
Frame_DF	View_control	Similar responsibilities as Frame_NPV (refers to Frame_NPV)

Frame_Para	View_control	create():
		responsible for the view of the parameter frame based on the input value list defined in the .xml file
		setpara():
		responsible for setting up the parameter

In the system responses column in Table 4.3, the integration mechanism, relationship detector and activator, proposed in the framework are described in detail.

For example, detecting related concepts, finding collaborative functions and activating the linkages between them. From the implementation point of view, they are identified as the responsibilities of the view-control objects and implemented via the following methods:

- getconcept()
- getrelatedconcept()
- getfunction()
- getrelatedfunction()
- activate()

Some of the code examples are presented in the prototype construction section.

Once the objects' behaviours and responsibilities have been identified, a detailed model can be created, which shows how objects interact with each other to provide automatic integration functionalities. Figure 4.8 is a partial sequence diagram for the Intelligent Tutoring Tools integration system (capital investment appraisal), which shows how the objects interact with each other over time.

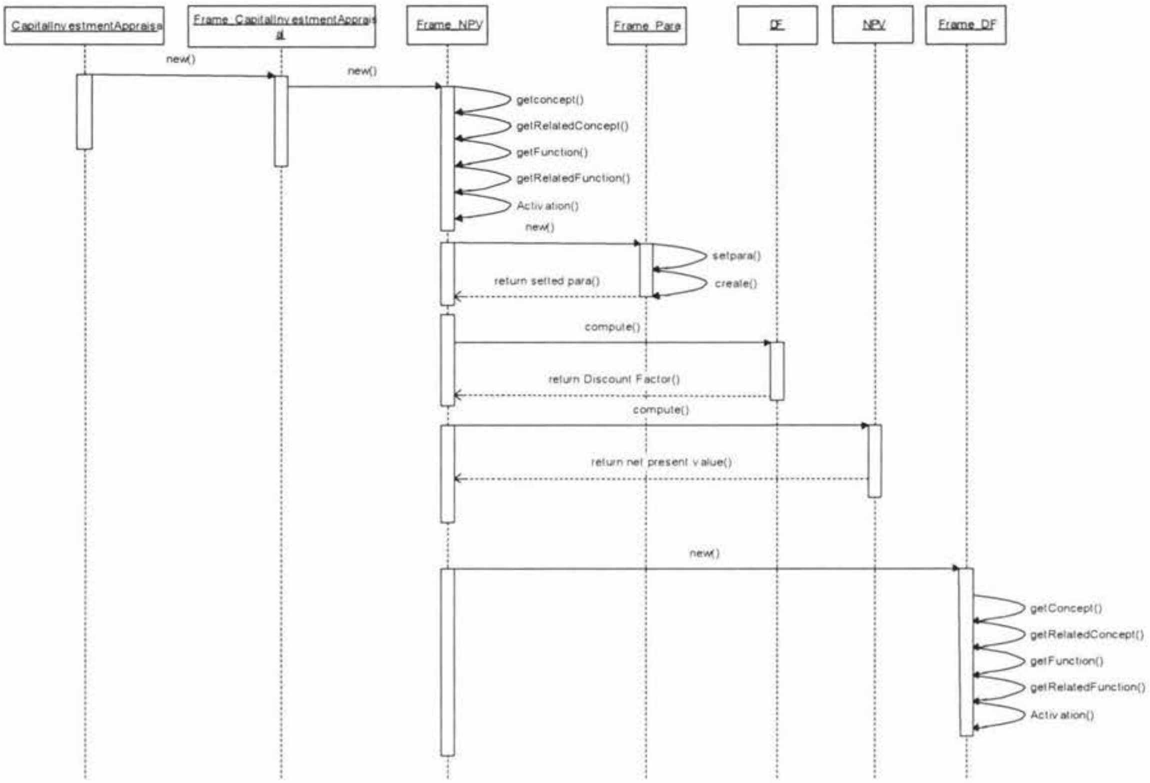


Figure 4.8. Partial sequence diagram for Intelligent Tutoring Tools integration system (capital investment appraisal)

Finally, in the system design stage, after designing the objects and their required interactions, the object model is refined to include the behaviours or implementation methods. Figure 4.9 is a partial view of the refined object class diagram for Intelligent Tutoring Tools integration system (capital investment appraisal).

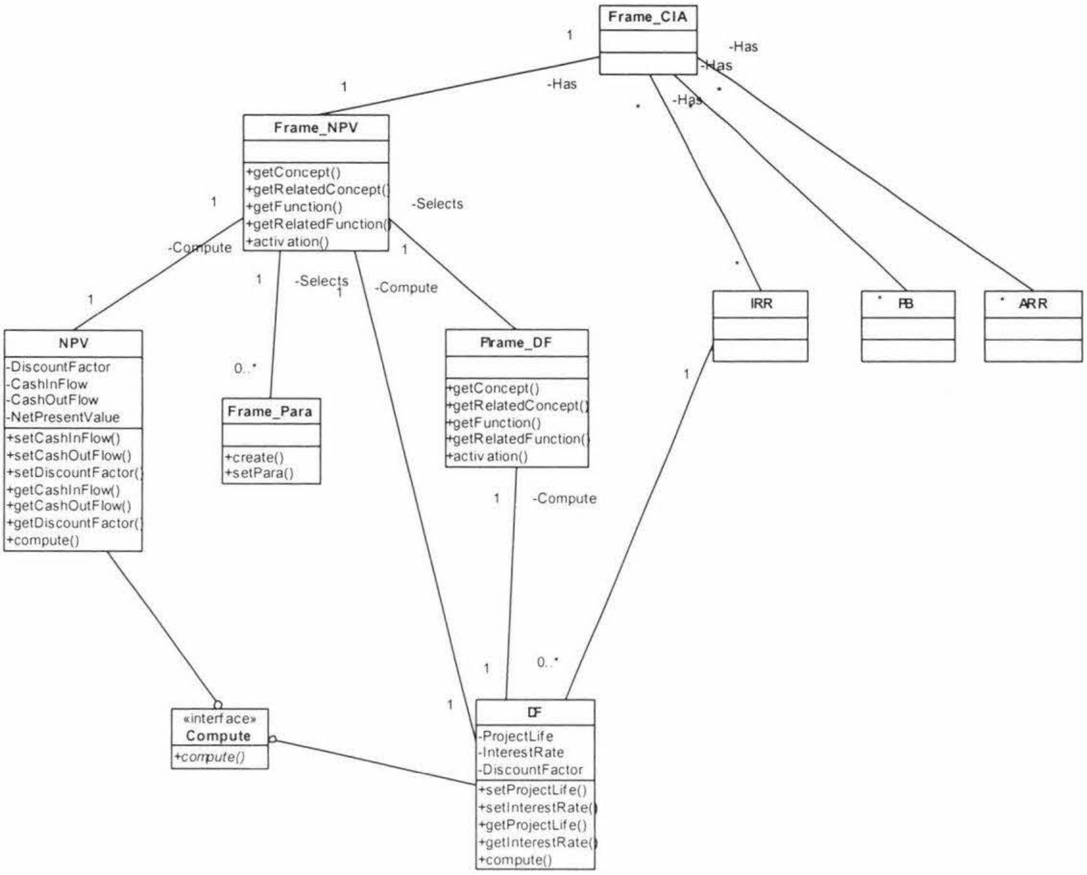


Figure 4.9. Partial object class diagram for Intelligent Tutoring Tools integration system (capital investment appraisal)

System design has refined the objects identified in the previous system analysis and introduced other objects. With these collaborative objects the system requirements are implemented to provide the learner an integrated learning system. The next section presents the implemented prototype system.

4.5 Construct Prototype

This section briefly introduces some technologies, JAVA and XML and a development tool, JBuilder, which are used in the prototype system development, followed by a metadata with XML binding example and some code samples for the relationship detection and activation. Finally, screenshots of the implemented system and a brief explanation are presented.

4.5.1 JAVA

JAVA was introduced by Sun Microsystems in 1995. Most programming languages need to be compiled or interpreted before running on the computer. The JAVA programming language is "unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called JAVA bytecodes —the platform-independent codes interpreted by the interpreter on the JAVA platform" (Sun Microsystems, Inc., 2003a). The unusual design principles, bytecode and JAVA Virtual Machine make JAVA platform-independent and "write once, run anywhere" possible. Since its first release, JAVA has grown rapidly in popularity and usage, because of its benefits of being "simple, object-oriented, distributed, interpreted, robust, secure, architecturally neutral, portable, and dynamic" (Sun Microsystems, Inc., 2003a).

There are three editions: JAVA 2 Platform Micro Edition (J2ME technology), JAVA 2 Platform, Standard Edition (J2SE technology), and the JAVA 2 Platform Enterprise Edition (J2EE technology).

- The J2ME specifically addresses the range of extremely tiny commodities such as smart cards or pagers.
- The J2SE platform is a fast and secure foundation for building and deploying client-side enterprise applications.
- The J2EE simplifies enterprise applications by basing them on standardized, modular and re-usable components Enterprise JavaBeans (EJB).

The J2SE is applicable to the prototype system development.

4.5.2 XML

Extensible Markup Language (XML) is a simple, very flexible text format derived from Standard Generalized Markup Language (SGML), a standard for how to create a document structure. XML is similar to the HyperText Markup Language (HTML). Both XML and HTML contain markup symbols to describe the contents of a page or file. HTML, however, describes the content of a web page only in terms of how it is to be displayed and interacted with, while XML describes the content in terms of what data is being described.

XML is a cross-platform, extensible and text-based standard for representing data. While it was "originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere" (W3C, 2003). XML is used in this prototype to bind the metadata describing the Intelligent Tutoring Tools' contents and interrelationships with other ITTs.

Furthermore, JAVA also provides support for the XML technology. It provides JAVA API for XML Processing (JAXP) which has functionalities for reading, manipulating, and generating XML documents (Sun Microsystems, Inc., 2003b).

4.5.3 JBuilder

Borland JBuilder is a widely used Integrated Development Environment (IDE) for JAVA. JBuilder consists of basic functionalities of the IDE, like language aware editing, project definition facilities, integrated compilation and stepwise debugger. It also offers Graphic User Interface (GUI) building and wizard functions. The latest version JBuilder X provides powerful functionalities to "speed EJB, Web, Web Services, XML, mobile, and database application development with two-way visual designers and rapid development to leading J2EE application servers" (Borland. Inc. 2004).

In this prototype system development, the utilisation of the functions and features provided by JBuilder, especially code edition, compilation, debugging, building graphical user interface, project management feature and XML supporting, greatly speeds up the prototype system development.

JBuilder Enterprise Version 7.0 is used in the system development.

4.5.4 Code Example

The first source code sample "getrelatedconcept" shows the system parses the information described in the XML file and detects the related concepts, which are defined under the category "Relation". The system then generates buttons to guide the learner to the related concepts (see Figure 4.10).

The second source code sample "activation" shows that the system compares the input set of one ITT with the output set of other ITTs. If any input value can be provided by another ITT an initially disabled button will be activated (see Figure 4.11).

An XML code sample shows that the XML binding metadata describes the Intelligent Tutoring Tools functions and interrelationships with other ITTs (see Figure 4.12). In the XML example, ITT Registry (which is used as a catalogue) does not exist. This is just an example of one type of registry system that may be used. In this prototype, the classname of the object is used as an identifier.

```
public void getrelatedconcept(){

  DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

    try{
        DocumentBuilder db = factory.newDocumentBuilder();
        Document doc = db.parse(urlString1); //urlString is the url to the xml file

        Element ele = doc.getDocumentElement();
        NodeList nl = ele.getElementsByTagName("relation");

        for(int i=0; i<nl.getLength(); i++){

            Element relation = (Element)nl.item(i);

            NodeList nlresource = relation.getElementsByTagName("resource");
            NodeList nldescription = elation.getElementsByTagName("description");

            for (int j=0; j < nlresource.getLength(); j++){
                try{
                  Node resource = nlresource.item(j);
                  String resource_string = resource.getFirstChild().getNodeValue();
                  Node description = nldescription.item(j);
                  String description_string = description.getFirstChild().getNodeValue();
                  concept_related_resource.add(resource_string);
                  concept_related_description.add(description_string);
                  jButton_concept_related[j].setBorder
                  (BorderFactory.createRaisedBevelBorder());
                  jButton_concept_related[j].setText(description_string);
                  jPanel1.add(jButton_concept_related[j], null);

                  if (j==0){
                   jButton_concept_related[j].addActionListener
                     (new java.awt.event.ActionListener() {
                     public void actionPerformed(ActionEvent e) {
                      jButton_concept_related0_actionPerformed(e);
                     }
                    });
                  }
::::::
```

Figure 4.10. Code sample of method “getrelatedconcept():”

```
public void activation(){

   try{
      for (int i = 0; i < input1.size(); i++){

      // get the input of NPV Learning Object
      String inputString = (String)input1.get(i);

      // find the corresponding output in other Learning Object
      int outputindex = output2.indexOf(inputString);

      // if any of the input of NPV object matches any output of other objects,
      // activate the linkage
      if (outputindex != -1){
       // System.out.println("index of input: " + i);
       // System.out.println(input1.get(i));
        if (i == 0) this.jButton_inflow.setEnabled(true);
        if (i == 1) this.jButton_outflow.setEnabled(true);
        if (i == 2) this.jButton_DiscountFactor.setEnabled(true);
        compareResult = true;
      }
    }
::::::
```

Figure 4.11. Code sample of method "activation():"

```
<itt>
     <general>
        <identifier>
          <catalog>ITT Registry</catalog>
            <entry>ittif.NPV</entry>
          </identifier>
        <description>
            <string language = “en”> Net Present Value </string>
        </description>
    </general>
     <relation>
          <kind>
            <source> LOMv1.0 </source>
            <value>isbasedon</value>
          </kind>
          <resource>
            <identifier>
                <catalog>ITT Registry</catalog>
                <entry>ittif.Frame_DF</entry>
            </identifier>
          <description>
                <string language=”en”> Discount Factor </string>
            </description>
          </resource>
     </relation>
    <relation>
     // relation description with Capital Investment Appraisal ITT
     </relation>
     <function id = "NPV">
          <funcdesc> calculate the Net Present Value </funcdesc>
          <output>
               <outname> NPV </outname>
               <outdatatype type = "float"></outdatatype>
               <outcomments> Net Present Value </outcomments>
          </output>
          <input>
               <inname> cashinflow </inname>
               <indatatype type = "float"></indatatype>
               <incomments> Cash Inflows </incomments>
          </input>
          <input>
               <inname> cashoutflow </inname>
               <indatatype type = "float"></indatatype>
               <incomments> Cash Outflows </incomments>
          </input>
          <input>
               <inname> discountfactor </inname>
               <indatatype type = "float"></indatatype>
               <incomments> Discount Factor </incomments>
          </input>
     </function>
</itt>
```

Figure 4.12. XML sample describing ITT functions and relationships

4.5.5 Screenshots of the Prototype System

In this section, the screenshots of the prototype system are presented. The brief explanation is from the design use case described in Section 4.4.2.

The Capital Investment Appraisal window is currently displayed on the screen waiting for the learner to select one of the four methods.

1. A learner selects NPV method.

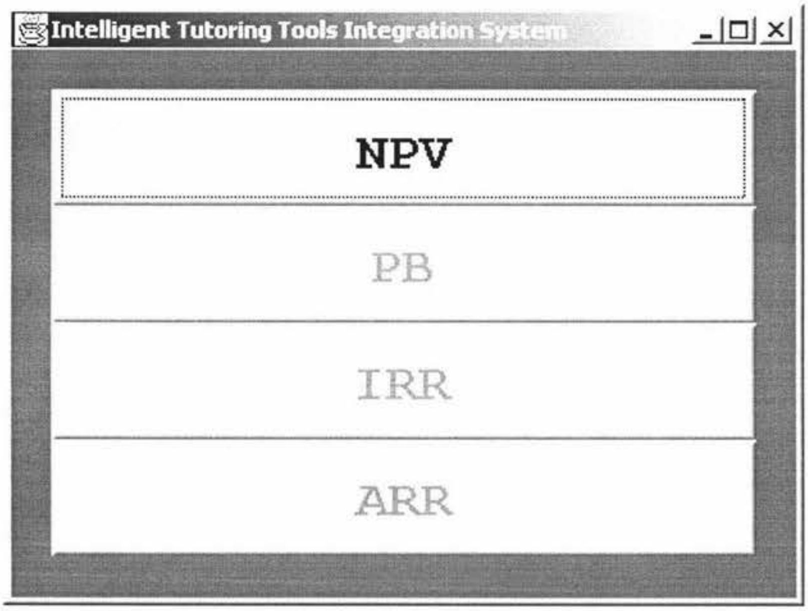
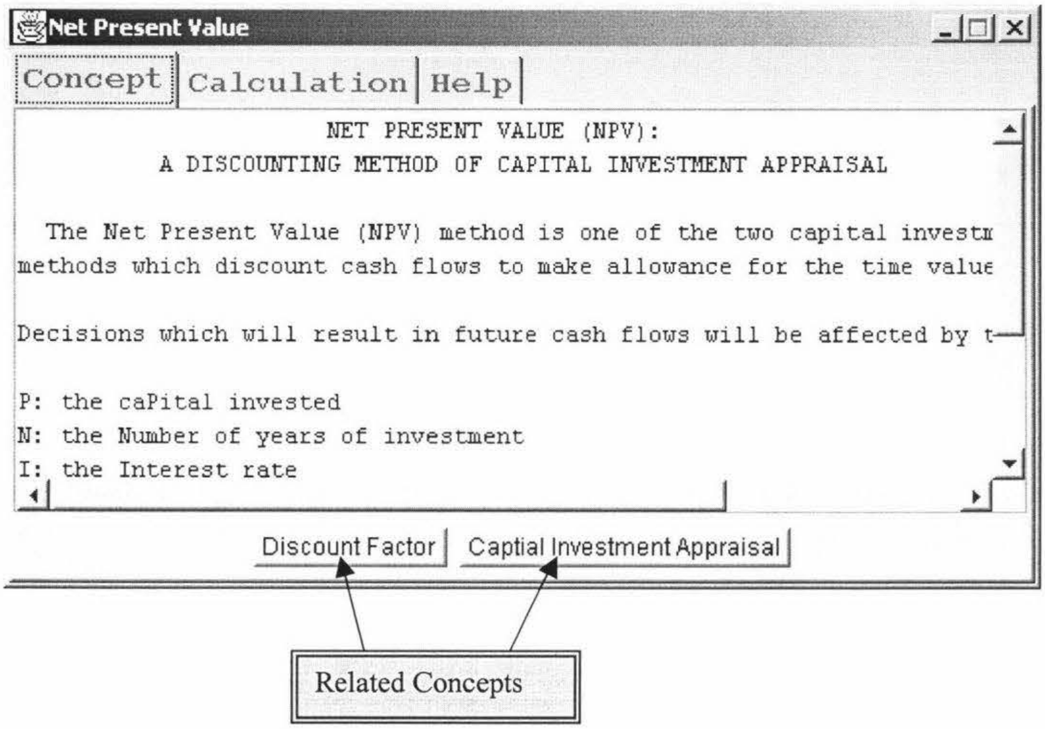


Figure 4.13. Capital investment appraisal window

2. The Intelligent Tutoring Tool "NPV" will create a screen to present the NPV concept. Meanwhile, it will check the related concepts. If there are some related concepts (Discount Factor & Capital Investment Appraisal), it will generate buttons to guide the user to the related concepts.



NET PRESENT VALUE (NPV):
A DISCOUNTING METHOD OF CAPITAL INVESTMENT APPRAISAL

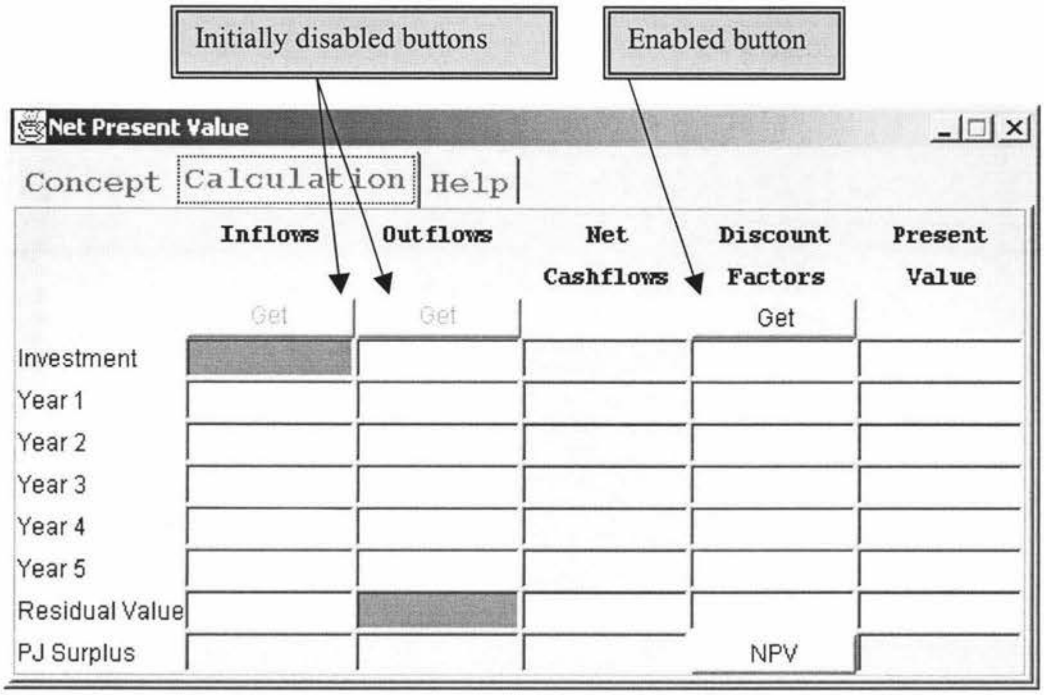
The Net Present Value (NPV) method is one of the two capital investm
methods which discount cash flows to make allowance for the time value

Decisions which will result in future cash flows will be affected by t

P: the caPital invested
N: the Number of years of investment
I: the Interest rate

Figure 4.14. Net present value concept window

3. The Intelligent Tutoring Tool "NPV" will also create a screen to present the NPV calculation. Meanwhile, it will check the input set of NPV as well as the output set of other ITTs. If any input value of NPV can be provided by another ITT, it will activate the button which allows the user to get that value from another ITT. For example, the Discount Factor can be provided by another ITT "DF", the system will then automatically activate the 'Get' button.



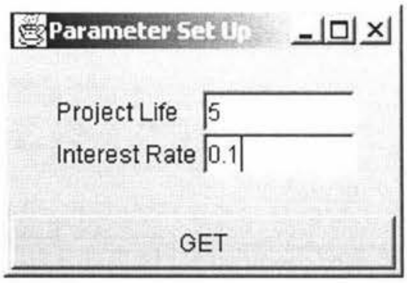
	Inflows	Outflows	Net Cashflows	Discount Factors	Present Value
	Get	Get		Get	
Investment					
Year 1					
Year 2					
Year 3					
Year 4					
Year 5					
Residual Value					
PJ Surplus				NPV	

Figure 4.15. Net present value calculation window

4. The learner selects "Get" discount factor from other ITT.

5. The system will generate a parameter screen based on the input set of ITT "DF", waiting for the learner's entry.

6. The learner enters the "project life" and "interest rate".

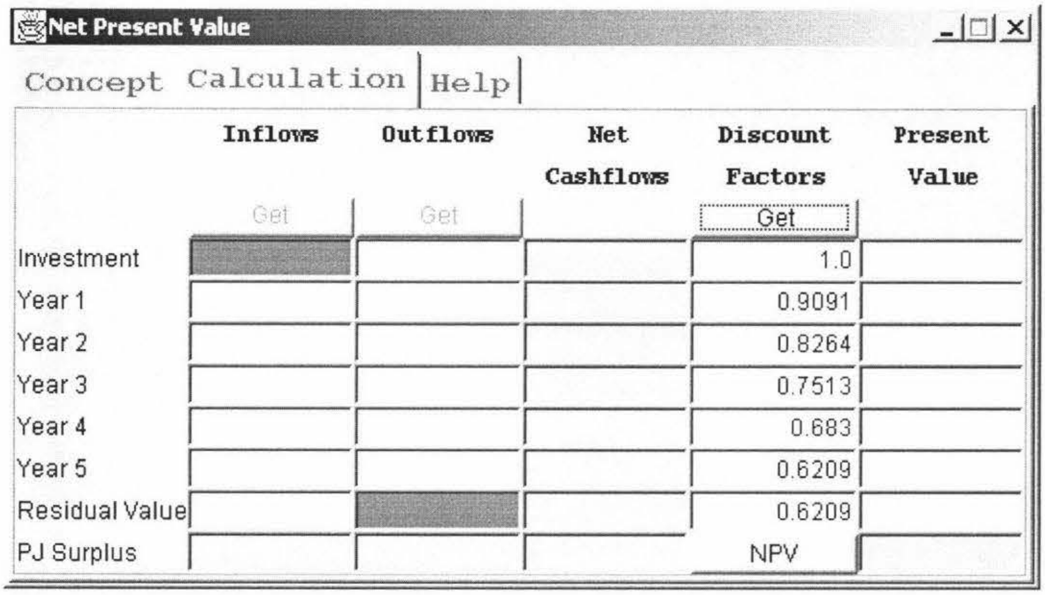


Project Life 5

Interest Rate 0.1

Figure 4.16. Parameter set up window

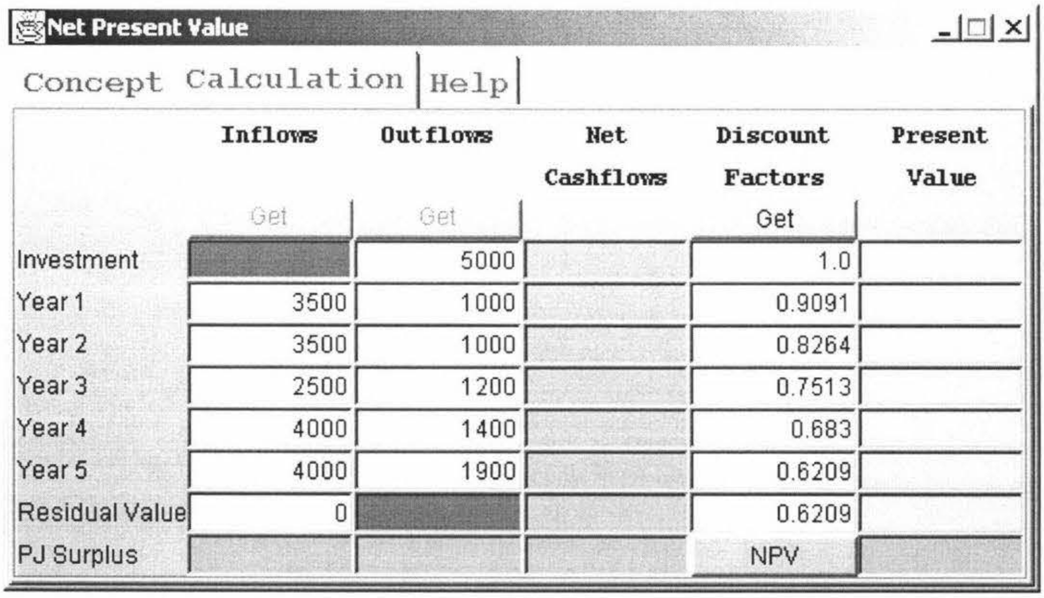
7. The system will get the input and pass to ITT "DF" for calculation; the returned value will be set in the NPV calculation screen.



	Inflows	Outflows	Net Cashflows	Discount Factors	Present Value
	Get	Get		Get	
Investment				1.0	
Year 1				0.9091	
Year 2				0.8264	
Year 3				0.7513	
Year 4				0.683	
Year 5				0.6209	
Residual Value				0.6209	
PJ Surplus				NPV	

Figure 4.17. Discount factor returned window

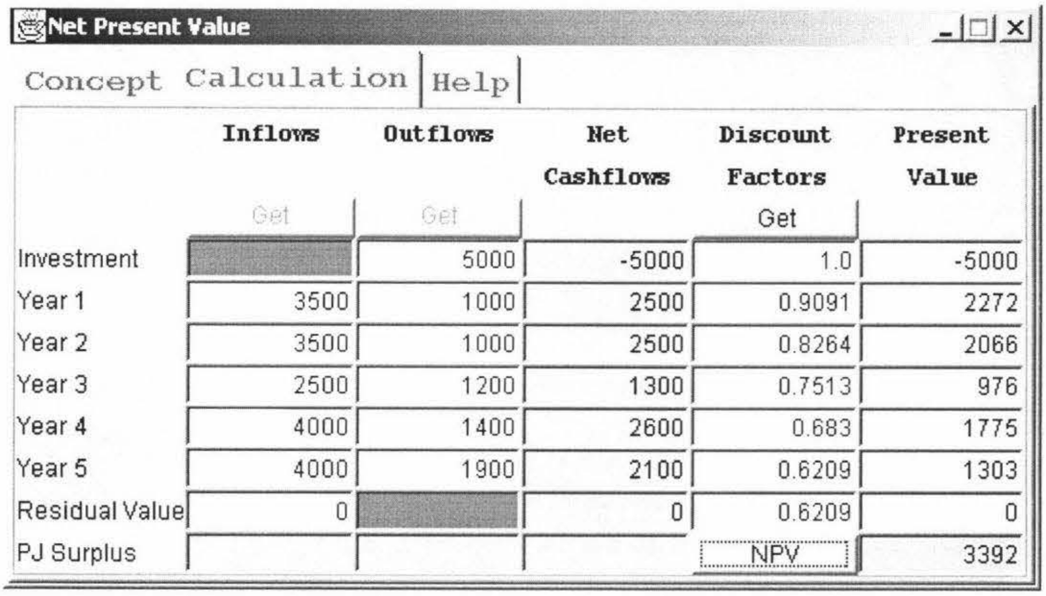
8. The learner enters other "cash inflows" and "cash outflows", and clicks NPV button.



	Inflows	Outflows	Net Cashflows	Discount Factors	Present Value
	Get	Get		Get	
Investment		5000		1.0	
Year 1	3500	1000		0.9091	
Year 2	3500	1000		0.8264	
Year 3	2500	1200		0.7513	
Year 4	4000	1400		0.683	
Year 5	4000	1900		0.6209	
Residual Value	0			0.6209	
PJ Surplus				NPV	

Figure 4.18. User entry for other information window

9. The system will get the input and pass to ITT "NPV" for calculation; the returned value will be set in the screen.

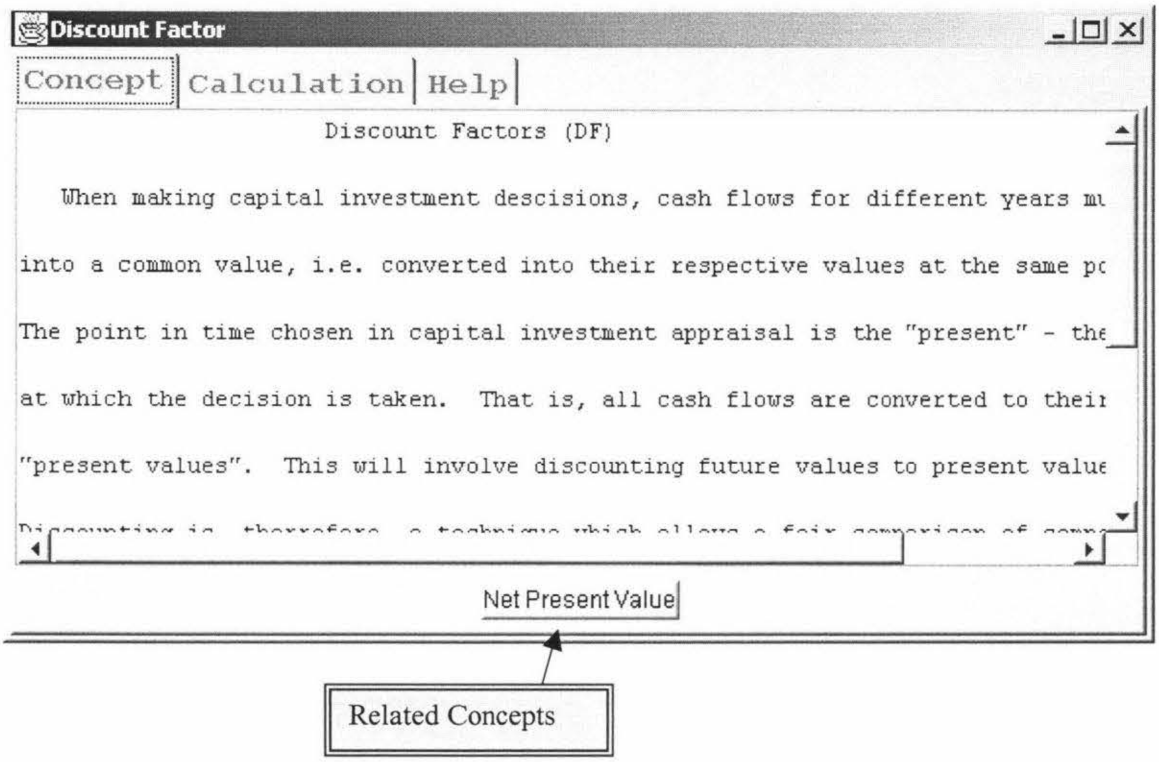


	Inflows	Outflows	Net Cashflows	Discount Factors	Present Value
	Get	Get		Get	
Investment		5000	-5000	1.0	-5000
Year 1	3500	1000	2500	0.9091	2272
Year 2	3500	1000	2500	0.8264	2066
Year 3	2500	1200	1300	0.7513	976
Year 4	4000	1400	2600	0.683	1775
Year 5	4000	1900	2100	0.6209	1303
Residual Value	0		0	0.6209	0
PJ Surplus				NPV	3392

Figure 4.19. Net present value result window

The above illustrates a typical course of events. Another routine is presented below:

At step 3, the learner selects the related concept "Discount Factor" from "Net Present Value" concept screen; the system will guide the learner to the related concept. When the learner finishes the study of concept "Discount Factor", he/she can go to related concept "Net Present Value" (Figure 4.20 and Figure 4.21).



Discount Factors (DF)

When making capital investment descisions, cash flows for different years mu

into a common value, i.e. converted into their respective values at the same po

The point in time chosen in capital investment appraisal is the "present" - the

at which the decision is taken. That is, all cash flows are converted to their

"present values". This will involve discounting future values to present value

Discounting is, therefore, a technique which allows a fair comparison of compa

Figure 4.20. Discount factor concept window

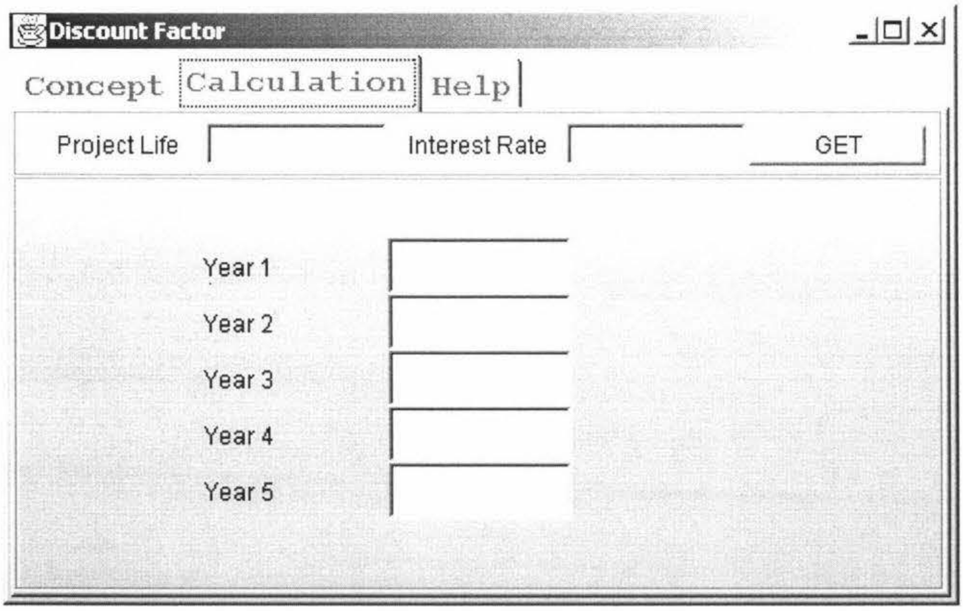


Figure 4.21. Discount factor calculation window

This section has presented the implemented prototype system; it is not a final and complete system, which means that the prototype "will not include the error checking, input data validation, security, and processing completeness of a finished application. Nor will it be as polished or offer the user help as in a final system" (Whitten, et al., 2002, p171). However, it quickly identifies the most crucial Intelligent Tutoring Tools integration requirements.

4.6 Summary

The objective of the prototype system, as described in Section 4.3, is to demonstrate that the application of the framework enables the system designer to identify the Learning Objects and the relationships among them to organize the Learning Objects in a pedagogical way and automate the integration process. However, the efficiency and effectiveness of the approach needs to be evaluated, and this is discussed in the next chapter.

Chapter 5

Formative Evaluation of the Prototype

A formative evaluation has been carried out for the Intelligent Tutoring Tools integration prototype. The feedback from the users was obtained through the use of questionnaires. The evaluation focused on the educational value of the integration of interrelated ITTs, such as enabling a higher-level intelligence, extending the scope of an individual ITT and collaborating with other ITTs to solve a more complex problem and contextualizing independent ITTs. It also aimed to assess the effectiveness of the prototype. Furthermore, user feedback was used to identify possible future improvements.

5.1 Participants of the Evaluation

There were a total of twenty-five participants, including:

- 2 Assistant Lecturers
- 2 Tutors
- 2 PhD Students
- 1 Junior Research Officer
- 4 Master students (one of them is Graduate Assistant)
- 4 Honours students (two of them are Graduate Assistants)
- 10 Undergraduate students

After working through the prototype, the participants were then asked to complete the questionnaire.

The next section summarizes and analyses the feedback received from the participants.

5.2 Evaluation Questionnaires and Summary

The questionnaire contained seven questions.

1. Are you familiar with any traditional intelligent tutoring system?

□ Yes □ No

16 out of 25 participants answered yes, the other participants commented that they were not familiar with traditional ITS. This response shows that on average the participants have some knowledge or experience with the traditional intelligent tutoring system, either as academic staff or students.

2. Do you think that the Learning Objects technology will practically improve the content reusability? Why?

□ Yes □ No

Comments:

All the participants thought that the Learning Objects technology would more or less improve the content reusability practically; they gave the following reasons for their answers:

- LO technology makes full use of the advantages of object-oriented approach.
- LO technology generates small content, which can be reused across different platforms or learning systems.
- It is always the case that sub-knowledge may be applicable in a related area.
- LO technology provides a standard framework to develop and index learning contents, which are shareable and retrievable.
- Learning Objects need to be reused in an instructional context to provide educational value.
- Learning Objects need to be designed and implemented in a standard way in order to improve the reusability.

3. Do you think that the Intelligent Tutoring Tool is also one type of Learning Objects? Why?

□ Yes □ No

Comments:

After an introduction to this project, all the participants thought that the Intelligent Tutoring Tool was also one type of Learning Objects. They gave the following reasons for their answers:

- ITT and LO use the same approach to achieve the content reusability.
- ITT and LO breakdown a complex concept or skills and focus on a small scope.
- ITT and LO contain instructional contents.

- ITT and LO are independent.
- ITT and LO are both described with metadata, they are retrievable and shareable.
- ITT needs to be developed and indexed according to a standard that others can reuse.
- ITT is a self-contained entity which carries out more intelligent tasks.

4. If the learner could focus on the current higher-level concept without repeating previously understood concepts, do you think this will help the learner master complex skills more efficiently?

	Strongly disagree				Strongly agree
	1	2	3	4	5
Answer:					
Comments:					

The average rating for this question was 3.52. The responses to this question varied, as the comments below illustrate:

- Significantly save the learner time.
- Prevent information overload.
- Reduce frustration of repeating lower-level concepts.
- Simplified user interface design.
- An overview or a "big picture" is necessary to put things into context.

- Reviewing the previously understood concepts is necessary and important in the learning process.
- The lower-level concept should be understood and mastered.

5. In this system, the learner is automatically presented with other interrelated ITTs for reference or further learning. Do you think this approach will benefit the learner?

	Strongly disagree				Strongly agree
	1	2	3	4	5
Answer:					
Comments:					

The average rating for this question was 4.16. The comments were given below:

- Students can learn at their own pace, more capable students can explore more complex concepts, and vice versa.
- Learners can learn ITT in a context.
- Learners are guided to the prerequisite concept or a higher-level concept.
- Automated learners have a great chance to learn something more.
- A sound navigation pattern is provided to the learner.
- Interrelated ITTs should be presented to the learner without information overload.
- Interrelated ITTs should be presented differently to skilled learners and beginners.

6. Did the prototype demonstrate how the interrelated ITTs are integrated and collaborated to solve a particular problem?

	Strongly disagree				Strongly agree
	1	2	3	4	5
Answer:					
Comments:					

The average rating for this question was 3.84. The comments were given below:

- The system demonstrates the integration between Net Present Value, Discount Factor and Capital Investment Appraisal well. It gives a clear picture of the relationship and collaboration between ITTs.
- It demonstrates clearly that one ITT has relationships with others. It would be better if more detail of the relationships could be shown (for example, what kind of relationship).
- The system demonstrates the independent ITTs in an integrated environment. If the system had demonstrated ITT in both segregated and integrated environments, it would have presented more clearly the benefits of the integration of Intelligent Tutoring Tools.

7. Based on your experience, what additional functionality will improve this prototype or integration of independent ITTs?

Comments:	

The comments given to this question are summarized below:

- ❖ If a few more ITTs could be integrated into the prototype, the benefits provided by integration may be more obvious.
- ❖ The prototype should show clearly the relationships (what kind) between ITTs.
- ❖ The presentation of concepts in the prototype could be improved with multiple formats, for example colours, headers and tool tips.
- ❖ The ITT in the prototype could be improved with the implementation of more tutoring functions.

5.3 Analysis of the Feedback

This section summarizes the evaluation in reference to:

(1) LO/ITT approach,

(2) Educational value of ITTs integration, and

(3) Effectiveness of the prototype developed.

(1) All of the participants agreed that the Learning Objects technology can improve the learning content reusability, largely due to its object approach and metadata standardisation. ITT can be considered a type of Learning Objects because ITT has a similar approach in breaking down a complex domain into small chunks of knowledge to achieve the reusability of learning contents. ITT is also indexed, shareable and retrievable. In addition, ITT has its own features carrying out more intelligent tasks

compared with other types of Learning Objects, such as learning contents in the basic forms (e.g. images, sound and text). One of the participants suggested that the development and description of ITT should be standardised.

(2) The overall rating for the educational value of integration ITT is positive. All of the participants agreed that this approach will provide learners with a better learning experience. The benefits include, self-paced learning, contextualized learning, and saving time. It was suggested that the presentation of interrelated concepts should be different according to the learner level and learning style, and that information overload should be prevented.

(3) Comments given by the participants show that the prototype developed gives a clear picture of the relationship and collaboration between ITTs. However, the relationships between ITTs could be grouped based on the relationship type, for example, "ispartof", "haspart", and "isbasisfor". This would make it easier for the learner to find a reference of the current learning ITT, or go on to a more advanced level.

Based on the evaluation and the points raised in the earlier chapters, further improvements can be attempted. These are discussed in the next chapter.

Chapter 6

Future Work and Conclusion

This chapter will discuss the possible improvements for the Intelligent Tutoring Tools integration framework and the prototype system. Finally, a conclusion will be presented.

6.1 Future work

The prototype development has completed its first iteration and achieved the pre-defined objectives. However, the framework development is an iterative, on-going process, and needs to be improved and refined along with the incremental system development cycle. Some possible future improvements, based on the evaluation for the first iteration and the points raised in the research, are addressed below.

6.1.1 Examination in a Real World Learning System

Chapter 3 presented a self-learning environment, in which users interaction with the system in the following ways: (1) learners can search and retrieve Learning Objects, (2) learning system designers can generate and public learning contents, and (3) system administrators can manage resources. The prototype focuses on the ITT type Learning Objects integration, which is the core functional requirement of a self-learning environment in terms of realizing the automation and contextualisation processes and providing better learning experiences. However, the integration framework needs to be implemented and examined in a real world learning system, for example, ITT type Learning Objects repository, to enable refinement and enhancement.

6.1.2 Extension of the Scope

The prototype is built in a narrow scope, namely the capital investment appraisal domain, which includes three ITTs: Discount Factor, Net Present Value and Capital Investment Appraisal. The framework is applied in the prototype development process to identify, organize, describe and network these ITTs. Its usability and implementability should be evaluated with a wider scope in real world learning environments.

6.1.3 Combination with the other ITT Features

The prototype also examines the Intelligent Tutoring Tools integration mechanism. The implemented ITTs are very simple and have limited functionalities. However, the three components, expert model, student model and tutor model, along with the random question generation and feedback features could be combined and implemented with the integration mechanism to provide a more intelligent entity. The combination could provide the learner with a better learning experience, for example, preventing an overload of interrelated concepts, based on the learner's capability and learning style.

6.1.4 Authoring and Metadata Mapping Tools

The generation of the content and the metadata of the Intelligent Tutoring Tools were done manually for this prototype. It was found to be quite a time-consuming and tedious task. An authoring tool for creating content and specifying the metadata would greatly enhance the usability of the integration framework.

Since there are various metadata standards, mappings between the different standards play important roles in terms of the improvement of the interoperability and reusability of the defined Intelligent Tutoring Tools.

6.2 Conclusions

This research has attempted to propose a framework to formalize the integration of Intelligent Tutoring Tools. The purpose of the integration of interrelated ITTs is to provide learners better learning in a self-learning environment. The integration of ITTs is a system-automated process and the advantages of the integration include:

- Enabling a higher-level intelligence
- Extending the scope of a single ITT

- ❖ Utilising the functions provided via other ITTs to solve a more complex problem, and providing related learning
- ❖ Putting interrelated ITTs into instructional contexts.

After reviewing and analysing integration mechanisms applied in the existing learning systems, a new ITT integration framework was proposed to extend and formalize the former ITT integration structures (Kinshuk & Patel, 1997, Kinshuk, et al. 2003) in two ways: identifying and organizing ITTs, and describing and networking ITTs. The Intelligent Tutoring Tools Integration Framework (ITTIF) has the following: (1) Ontology to set up an explicit conceptualisation in a particular domain, (2) Object Design and Sequence Theory to identify and arrange Learning Objects in a pedagogical way, (3) Metadata to describe the identified ITTs and their interrelationships in a cross-platform XML format, and (4) Integration Mechanism to detect and activate the contextual relationship.

The proposed framework was then applied and examined with an implementation of a ITTs integration system (prototype) in a specified capital investment appraisal domain. The Prototype development has completed its first iteration, going through the following development phases: an initial system analysis, definition of the objective, specification of the prototype scope, construction and evaluation.

During the prototype system development, the application of the framework has provided a standardised approach, which proved able to help learning system designers to identify and organize Intelligent Tutoring Tools in a pedagogical way. Furthermore, Learning Object metadata (Relation Category) was applied to describe the relationships among identified ITTs, some metadata extensions were also identified for describing the functionality. Finally, an Integration Mechanism was formalised to carry out the core intelligent tasks, for example, detecting related concepts, finding collaborative functions and activating the linkages between them.

In the final stage of the prototype development, formative evaluation, all of the participants commented that the prototype clearly presented the integration of ITTs, and agreed that the integration of ITTs can provide learners with a better learning experience.

References:

ADL. (2003a). Advanced Distributed Learning initiative website, Available: http://www.adlnet.org/

ADL. (2001a). Advanced Distributed Learning initiative: The SCORM Overview Version 1.2, Retrieved October 10, 2003, from: http://www.adlnet.org/

ADL. (2003b). Advanced Distributed Learning initiative: The SCORM Content Aggregation Model Version 1.3 Working Draft 1, Available: http://www.adlnet.org/

ADL. (2003c) Advanced Distribute Learning initiative: The SCORM Run-Time Environment Version 1.3 Working Draft 1, Available: http://www.adlnet.org/

ADL. (2003d) Advanced Distribute Learning initiative: The SCORM Sequencing and Navigation Version 1.3 Working Draft 1, Available: http://www.adlnet.org/

ALI. (2003) Apple Learning Interchange website. Retrieved July 21, 2003, from http://ali.apple.com/edres

ASTD. (2003) American Society for Training & Development: online magazine all about e-learning. Available: http://www.learningcircuits.org/

Barron T. (2000) Learning Object Pioneers [On-Line] Available: http://www.learningcircuits.org/mar2000/barron.html

Bennett, S., Mcrobb, S. & Farmer, R. (2002) Object-Oriented Systems Analysis and Design Using UML (2nd Edition). Maidenhead Berkshire: McGraw Hill.

Borland. Inc. (2004) JBuilder: the leading development solution for Java. Available: http://www.borland.com/jbuilder/

Boyle, T. (2003). Design principles for authoring dynamic, reusable learning objects. Australian Journal of Educational Technology, 19(1), 46-58. Available: http://www.ascilite.org.au/ajet/ajet19/boyle.html

Bradley, Boyle (2003) The Development and deployment of multimedia learning objects ED_MEDIA 2003 World Conference on Educational Multimedia, Hypermedia & Telecommunications Learning Objects 2003 Symposium: Lessons learned, Questions Asked June 24, 2003; Honolulu, Hawaii, USA

Brennan, M., Funke, S., Anderson, C. (2001) "The Learning Content Management System: A new eLearning Market Segment Emerges." IDC White Paper Published May 2001. Available: http://www.idc.com/

Cisco System. (2001) Reusable Learning Object Strategy: Designing Information and Learning Objects Through Concept, Fact, Procedure, Process, and Principle Templates. Retrieved 17 May 2003 from: http://business.cisco.com/servletwl3/FileDownloader/iqprd/86575/86575_kbns.pdf

Collins, D. (1995) Designing Object-Oriented User Interfaces. Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc.

DCMI. (2003) Dublin Core Metadata Initiative website. Available: http://www.dublincore.org/

Donello J. F. (2002) Theory & Practice: Learning Content Management Systems Retrived from: http://www.elearningmag.com/

Duncan, C. (2001) Why Digital Repositories, An intrallect white paper, Retrieved from: http://www.intrallect.com/papers/why_digital_repositories.pdf

EdNA. (2003) Education Network Australia website. Available: http://www.edna.edu.au/
Ellis, R.K. (2001) LCMS Roundup. Retrieved 28 September 2003 from: http://www.learningcircuits.org/2001/aug2001/ttools.html

EOE. (2003) Education Objects Economy website. Retrieved 20 May 2003 from: http://www.eoe.org/FMPro?-db=portals.fp3&-token=home&-format=agindex.htm&-script=UpdateFrontPage&-Findall

Friesen, N. (2003) Three Objections to Learning Objects Retrieved 20 October 2003 from: http://phenom.educ.ualberta.ca/~nfriesen/

GEM (2003a) The Gateway to Educational Materials website. Available: http://www.thegateway.org/

GEM (2003b) GEM Project Site. Available: http://www.geminfo.org/

Gibbons, A. S., Nelson, J. & Richards, R. (2000). The nature and origin of instructional objects. In D. A. Wiley (Ed.), The Instructional Use of Learning Objects: Online Version. Retrieved from http://reusability.org/read/chapters/gibbons.doc

Gong, X. (2002) Web-course Search Engine, Master thesis, Messay University, Palmerston North, New Zealand

Greenberg, L. (2002) LMS and LCMS: What's the difference? Retrieved 02 October 2003 from: http://www.learningcircuits.org/2002/dec2002/greenberg.htm

Gruber, TR. (1993) A Translation Approach to Portable Ontology Specifications

Knowledge Systems Laboratory Technical Report KSL 9-71. Retrieved 20 October 2003 from: http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html

Heery, R., Patel, M. (2000) Application profiles: mixing and matching metadata schemas. Retrieved 23 November 2003 from: http://www.ariadne.co.uk/issue25/app-profiles/intro.html

Hodgins, H. W. (2000a) Into the future. Retrieved 19 April 2003. from http://www.learnativity.com/download/MP7.PDF

Hodgins, H. W. (2000b). The future of learning objects. In D. A. Wiley (Ed.), The Instructional Use of Learning Objects: Online Version. Retrieved 15, October, 2003, from http://reusability.org/read/chapters/hodgins.doc

Ielearning (2003) Integrity eLearning Inc. website. Retrieved 16 October 2003, from: http://www.ielearning.com/wbt/features/whatisanLMS.cfm

ILumina. (2003) iLumina website. Available: http://turing.bear.uncw.edu/iLumina/index.asp

Jacobsen, P. (2002) LMS vs. LCMS: Take advantage of the differences to save time and money. Retrieved 10 October 2003 from http://www.elearningmag.com/

Kinshuk & Patel A. (1997). A Conceptual Framework for Internet based Intelligent Tutoring Systems. Knowledge Transfer (volume II) (Ed. A. Behrooz), pAce, London, 117-124 (ISBN 1-900427-015-X).

Kinshuk, Patel A. & Russell D. (1999) Hyper-ITS: A Web-based Architecture for Evolving a Configurable Learning Environment. Staff and Educational Development International Journal, 3 (3), 265-280 (ISSN 0971-9008).

Kinshuk, Shi, L., Patel A. (2003) Higher level intelligence through horizontal an vertical newtworking of tutoring applications. 10th International Conference on Telecommunications ICT'2003 Conference Proceedings, February 23 - March 1, (pp.1735-1740). Piscataway, NJ: IEEE.

L'Allier, J. (1998). NETg's Precision Skilling The Linking of Occupational Skills Descriptors to Training Interventions. Retrieved 26 September 2003 from: http://www.netg.com/research/whitepapers/pskillpaper.asp

Larman C. (2002). Applying UML and Patterns. Prentice Hall PTR. Prentice-Hall, Inc. Upper Saddle River. NJ

LOM. (2002). Draft standard for Learning Object Metadata. IEEE 1484.12.1-2002 15 July 2002. Retrieved 20 August 2003 from: http://ltsc.ieee.org/doc/wg12/

LTSC. (2003) Learning Technology Standards Committee of the IEEE website. Available: http://ltsc.ieee.org/

MERLOT (2003): Multimedia Educational Resources for Learning and Online Teaching website. Available: http://merlot.org/

Merrill M. David (1998). Knowledge Objects. CBT Solutions March/April. 1-11 Available: http://www.id2.usu.edu/Papers/KnowledgeObjects.PDF

Millea J. (2003) The EdNA Metadata Standard. Retrieved 18 September 2003 from: http://www.educationau.edu.au/

Mohan, P., Brooks, C. (2003) Learning Objects on the Semantic Web, The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03) July 09 - 11, 2003 Athens, Greece

Neven, R., Duval, E. (2002) Reusable Learning Objects: a Survey of LOM-Based Repositories, Proceedings of the 10th ACM International Conference on Multimedia 2002, December 1-6, 2002, Juan les Pins, France. ACM, 2002, ISBN 1-58113-620-X. 291-294

NNDSB. (2003) Near North District School Board website. Dewey Decimal Classification System, Available: http://www-lib.nearnorth.edu.on.ca/dewey/ddc.htm

OCLC (2003) Online Computer Library Center: Introduction to Dewey Decimal Classification. Available http://www.oclc.org/dewey/

Patel A. & Kinshuk (1997a) Intelligent Tutoring Tools on the Internet - extending the scope of distance education. 18th ICDE World Conference, June 2-6, 1997, State College, PA, USA

Patel A. & Kinshuk (1997b) Intelligent Tutoring Tools in a Computer Integrated Learning Environment for introductory numeric disciplines. Innovations in Education and Training International Journal, 34 (3), August, 200-207 (ISSN 1355-8005).

Reigeluth, C.M. & Nelson, L.M. (1997). A new paradigm of ISD? In R.C. Branch & B.B. Minor(Editors), Educational media and technology yearbook (Vol.22, pp. 24-35). Englewood, CO: Libraries Unlimited

Rengarajan, R. (2001) LCMS and LMS Taking Advantage of Tight Integration. Retrieved 15 October 2003 from: http://www.e-learn.cz/soubory/lcms_and_lms.pdf

Shepherd, C. (2000) "Objects of Interest," TACTIX, December 2000. Published in IT Training, February 2001 Retrieved 13 July 2003, from: http://www.fastrak-consulting.co.uk/tactix/features/objects/objects.htm#Objects%20defined

Shute, V.J. & Psotka, J. Intelligent Tutoring Systems: Past, Present, And Future (2002) Retrieved 12 October, 2003, from http://www.aect.org/intranet/publications/edtech/pdf/19.pdf

Sun Microsystems, Inc. (2003a). About the java technology. Retrieved 12 December 2003 from: http://java.sun.com/

Sun Microsystems, Inc. (2003b). Java API for XML processing (JAXP). Retrieved 12 December 2003 from: http://java.sun.com/xml/jaxp/index.html

Sutton , S. A. & Mason, J. (2001) The Dublin Core and Metadata for Educational Resources, DC-2001 Proceedings of the International Conference on Dublin Core and Metadata Applications 2001, Tokyo, Japan. Available: http://www.informatik.uni-trier.de/~ley/db/conf/dc/dc2001.html

TeleCapmus. (2003), TeleCapmus website. Retrieved 20 September 2003 from: http://courses.telecampus.edu/about/index.cfm?fuseaction=introduction

Thomas, D. W. (n.d.) "Dr. Tom's metadata Guide". Retrieved 12 July 2003 from: http:// www.imsproject.org/drtommeta.html

van Merriënboer, J.J.G., Clark, R.E. and de Croock, M. B. M. (2002) Blueprints for Complex Learning: The 4C/ID-Model. Educational Technology Research and Development, 50(2), pp. 39-64

W3C (2003) Extensible Markup Language (XML) Available: http://www.w3.org/XML/

Wagner. E. O. (2002) The new Frontier of learning object design The eLearning Developers' Journal 18 June 2002. Available: http://www.elearningguild.com/

Warren, L. (2000). A Primer on Learning Objects. Retrieved 5 August 2003 from: www.learningcircuits.org/mar2000/primer.html

Whatis.com (2003) Whatis.com website. Available: http://whatis.techtarget.com/

Whitten, J.L., Bentley, L.D., & Dittman, K.C. (2001) System Analysis and Design Methods (5^{th} Edition). New York: McGraw Hill.

Wiley, D. A. (1999) "The Post-LEGO Learning Object," November 5, 1999 Retrieved from: http://wiley.ed.usu.edu/docs/post-lego/

Wiley, D. A. (2000) Learning Object Design And Sequencing Theory, Phd Dissertation, Brigham Young University, USA

Wiley, D. A. (2001) "Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy," in D. A. Wiley, ed., The Instructional Use of Learning Objects: Online Version. Retrieved Oct, 2003 from http://reusability.org/read/chapters/wiley.doc

Wiley, D. A. (2003) Learning Objects: Difficulties and Opportunities, Retrieved Dec, 2003 from: http://wiley.ed.usu.edu/articles.html

Wilkinson-Riddle, G., & Patel, A., (Eds.). (1997). Interactive Accounting: The Byzantium Workbook. Massachusetts, Malden: Blackwell Publishers.