ISSN:0975-9646
S. Vasanthi et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (2) , 2012,3337 - 3340

Learning objects- an object oriented analysis, design model and research review

S. Vasanthi [1], M.K. Jayanthi [2]

School of Computer Science and Engineering
VIT University, Vellore, Tamil Nadu, India
[1]vasanthi.s@vit.ac.in
[2]jayanthi.mk@vit.ac.in

***Abstract*— This paper is devoted to mobile agents and their use for E-learning resource management. We provide a brief overview. The overview introduces the concept of mobile agent, enumerates the claimed benefits, and discusses the state of the art of application in this domain. With the rapid evolution of Internet, information overloading is becoming a common phenomenon, it is necessary to have a tool to help user to extract useful information from Internet. E-learning applications also face similar problem. At present, commercialized e-learning systems lack information search tools to help user search for the course information, and none of them have explored the power of mobile agent. Mobile agent is a suitable tool particularly for Internet information retrieval.**

***Keywords*— Object oriented system development, E-Learning, Mobile agent, Object oriented design.**

I. INTRODUCTION

Software development is dynamic and always undergoing major change. Today a vast number of tools and methodologies are available for system development. System development refers to all activities that go into producing information system solution. System development activities consist of system analysis, modeling, design, implementation, testing and maintenance. A software development methodology is series of processes that, if followed, can lead to the development of an application. Software engineering has traditionally been an expensive and time-intensive process. Object-oriented analysis and design is the principal industry-proven methodology that answers the call for a more cost-effective, faster way to develop software and systems. Object-oriented technology cuts development time and overhead, leading to faster time to market and significant competitive advantage, enabling software engineers to produce more flexible and easily maintainable applications

II. LEARNING OBJECT

A learning object is "a collection of content items, practice items, and assessment items that are combined based on a single learning objective" [1]. The term is credited to Wayne Hogins when he created a working group in 1994 bearing the name [2] though the concept was first described by Gerard in 1967[3]. Learning objects go by many names, including content objects, chunks, educational objects, information objects, intelligent objects, knowledge bits, knowledge objects, learning components, media objects, reusable curriculum components, nuggets, reusable information objects, reusable learning objects, testable reusable units of cognition, training components, and units of learning.

Learning objects offer a new conceptualization of the learning process: rather than the traditional "several hour chunk", they provide smaller, self-contained, re-usable units of learning [4]. They will typically have a number of different components, which range from descriptive data to information about rights and educational level. At their core, however, will be instructional content, practice, and assessment. A key issue is the use of metadata. Learning object design raises issues of portability, and of the object's relation to a broader learning management system.

III. OBJECT ORIENTED SYSTEM DEVELOPMENT METHODOLOGY

OO development offers a different model from the traditional software development approach. This is based on functions and procedures. To develop s/w by building self contained modules or objects that can be easily replaced, modified and reused. In OO environment, s/w is a collection of discrete object that encapsulate their data as well as the functionality to model real world ―objects. Each object has attributes (data) and method (function). Objects grouped in to classes and objects are responsible for it. A chart object is responsible for things like maintaining its data and labels and even for drawing itself.

A. Benefits of Object Orientation

Benefits are faster development, reusability and increased quality. Object technology emphasizes modeling the real world and provides us with the stronger equivalence of the real world‘s entities (objects) than other methodologies. Raising the level of abstraction to the point where application can be implemented in the same terms as they are described**.**

B. Why object orientation?

To create sets of objects that work together concurrently to produce s/w that better, model their problem domain that similarly system produced by traditional techniques. It adapts to changing requirements, Easier to maintain, More robust, Promote greater design code reuse, Higher level of abstraction, Seamless transition among different phases of software development, Encouragement of good programming techniques and promotion of reusability.

C. What is an object?

The term object was first formally utilized in the Similar language to simulate some aspect of reality. An object is an entity. It knows things (has attributes) and it does things (provides services or has methods)

D. Object

In an object-oriented system, everything is an object: numbers, arrays, records, fields, files, forms, an invoice, etc. An Object is anything, real or abstract, about which we store data and those methods that manipulate the data. Conceptually, each object is responsible for itself. A window object is responsible for things like opening, sizing, and closing itself. A chart object is responsible for things like maintaining its data and labels, and even for drawing itself. Two Basic Questions When developing an O-O application, two basic questions always arise. What objects does the application need? What functionality should those objects have?

E. Traditional Approach

The traditional approach to software development tends toward writing a lot of codes to do all the things that have to be done. You are the only active entity and the code is just basically a lot of building materials.

F. Object-Oriented Approach:

OO approach is more like creating a lot of helpers that take on an active role, a spirit, that form a community whose interactions become the application.

G. Object's Attributes

Attributes represented by data type. They describe objects states. In the Car example the car's attributes are: Colour, manufacturer, cost, owner, model, etc.

H. Object's Methods

Methods define objects behaviour and specify the way in which an Object's data are manipulated. In the Car example the car's methods are: Drive it, lock it, tow it, carry passenger in it.

I. Objects are grouped in classes

The role of a class is to define the attributes and methods (the state and behavior) of its instances. The class car, for example, defines the property color. Each individual car (object) will have a value for this property, such as "maroon," "yellow" or "white."

J. Class Hierarchy

An object-oriented system organizes classes into subclass-super hierarchy. At the top of the hierarchy are the most general classes and at the bottom are the most specific. A subclass inherits all of the properties and methods (procedures) defined in its super class.

K. Inheritance (programming by extension)

Inheritance is a relationship between classes where one class is the parent class of another (derived) class. Inheritance allows classes to share and reuse behaviours and attributes. The real advantage of inheritance is that we can build upon what we already have and, reuse what we already have.

L. Multiple Inheritances

OO systems permit a class to inherit from more than one super class. This kind of inheritance is referred to as multiple inheritances.

M. Object Oriented Systems Development Life Cycle

Goals: The software development process, Building high-quality software, Object-oriented systems development, Use-case driven systems development, Prototyping, Rapid application development, Component-based development, Continuous testing and reusability.

N. Software Process:

The essence of the software process is the transformation of users needs to the application domain into a software solution.

O. Software Quality

There are two basic approaches to systems testing. We can test a system according to how it has been built. Alternatively, we can test the system with respect to what it should do.

P. Quality Measures

Systems can be evaluated in terms of four quality measures: Correspondence, Correctness, Verification, and Validation. Correspondence measures how well the delivered system corresponds to the needs of the operational environment. It cannot be determined until the system is in place. Correctness measures the consistency of the product requirements with respect to the design specification. Verification is to predict the correctness. Validation is to predict the correspondence**.**

IV. Object-Oriented Systems Development activities

Object-oriented analysis, Object-oriented design, Prototyping, Component-based development, Incremental testing.

A. Object-Oriented Analysis

OO analysis concerns with determining the system requirements and identifying classes and their relationships that makes up an application.

B. Object-Oriented Design

The goal of object-oriented design (OOD) is to design the classes identified during the analysis phase, the user interface and Data access.

C. OOD activities include

Design and refine classes, Design and refine attributes, Design and refine methods, Design and refine structures, Design and refine associations, Design User Interface or View layer classes, Design data Access Layer classes.

D. Prototyping

A Prototype enables you to fully understand how easy or difficult it will be to implement some of the features of the system. It can also give users a chance to comment on the usability and usefulness of the design.

E. Component-based development (CBD)

CBD is an industrialized approach to the software development process. Application development moves from custom development to assembly of pre-built, pre-tested, reusable software components that operate with each other.

F. Rapid Application Development (RAD)

RAD is a set of tools and techniques that can be used to build an application faster than typically possible with traditional methods. RAD does not replace SDLC but complements it, since it focuses more on process description and can be combined perfectly with the object-oriented approach.

G. Incremental Testing

Software development and all of its activities including testing are an iterative process. If you wait until after development to test an application for bugs and performance, you could be wasting thousands of dollars and hours of time.

V. MOBILE AGENTS AND E-LEARNING RESOURCE MANAGEMENT

Mobile agents have raised considerable interest in the research community. In the past few years several platforms have been developed and mobile agents have been used in applications ranging from network resource management to automatic software distribution. Several benefits are associated with their use and the successful applications in electronic commerce are given to back the claimed benefits.

The primary characteristics of mobile agent are mobility and autonomy. The mobility borrows a lot from process migration which consists of transferring a process from one computer to another. The code, the data, and the running state of the mobile agent are all moved to the destination when migration occurs. The autonomy also gives mobile agent some artificial intelligence features. Mobile agent not only decides what to do next according to its autonomous strategy, but also can change its autonomous strategy to fit in with the new situation that some external changes cause.

Mobile agent platform is the special environment for mobile agents' execution. It is also called MAE (Mobile Agent Environment).

A. Agent model

This model gives an internal structure of the mobile agent in the platform, indicating how to build a mobile agent that can act on behalf of another entity.

B. Autonomy model

This model indicates how to handle the internal messages to manage the mobile agents in the platform.

C. Migration model

This model presents how a mobile agent moves to another platform in different physical nodes.

D. Communication model

Communication model cares how the mobile agents in the platform communicate with the external world.

E. Naming and locating model

Every mobile agent in the platform must have a unique name to be identified. The model cares how to name the mobile agent and how to locate it form the name.

F. Security model

Security model not only protects mobile agents against the platform, but also protects the platform against mobile agents.

G. Service locating model

Some mobile agents can provide some services in the internet. This model provides the location and the interface of the services. In the past few years, a lot of mobile agent platforms have been built. But most of them are implemented as Java applications. Famous platforms commonly used today include Aglets [5], Grasshopper [6], and Voyager [7]. The prototypes here are several benefits which are associated with mobile agents: [8].

H. Concurrency

A lot of mobile agents can achieve one goal at the same time. Each mobile agent has its own process.

I. Asynchrony

Mobile agents can migrate between physical nodes autonomously to finish the work. If there is no synchronous requirement, the result can be an asynchronous feedback. User need not trace the process and wait for result.

J. Reduce network usage

Mobile agents can continue roaming and working on behalf of users even when users are disconnected. User need not connect to the internet all the time.

K. Better performance

Compared to the number and length of messages exchanged in a traditional client/server application, performance is improved when the mobile agent's transferred code is small in size.

L. Dynamic interface upgrades

Mobile agents can upgrade their interfaces easily when they are running, while the client/server approaches are too static to adapt to rapid changes. Although mobile agents have several advantages, they have so far been used in experimental settings. In our opinion, deployment has been hindered by the security problem. Although a lot of researches have been given on mobile agent security, the problem has not yet been solved in a satisfactory manner.

M. Benefits from Mobile Agents

We use a lot of mobile agents in the process of sharing the resource, and the expected benefits from mobile agents are briefly discussed below: Mobile agents can improve performance over the client/server approach, especially in the situation that the registration data and agent's code are small in size. Mobile agents can react to the changeable situations to protect the learning resource. So the whole system will be more robust. Mobile agents can work locally. It is more convenient to cooperate with the internal subsystems of the platform. Mobile agents can compare and select the suitable resources to the student, during travelling around online. Mobile agents can give an asynchronous feedback. Student can disconnect to the network while the agent is working, instead of connecting to the network to wait the result. The agent can return the learning resources back to the student when he connects to the network later.

VI. CONCLUSION

Although the mobile agents have a good performance in the E-learning resource management, the resources protected by the REL are very safe, the agents who carrying them are not robustness. The result is that the legal user may not acquaint the resources for the weak protection of mobile agents. The second lesson is that the cost of an intelligent mobile agent is too high. The cost comes from two factors. One is the intelligent design, which takes a long time to make agent well-defined and adds a lot of codes to the agent for handling strategies. The other is migration, which gives convenience to communication but costs a lot of time for reloading the agent, especially the heavy agent.

In spite of these, the benefits from mobile agents show that using mobile agents to manage E-learning resources is a desirable way. We will continue our work to form a more perfect E-learning resources management system in the future and add more intelligent mobile agents to the platform. An extended KQML (Knowledge Query and Manipulation Language) will be used in the future to reduce the cost of migration also. And we will try to make the agent data much safer. Our ultimate goal is that using all mobile agents' advantages to improve the performance of E-learning resources management, building a robust intelligent platform.

ACKNOWLEDGMENT

The authors would like to thank VIT authorities for providing facilities and opportunity to carry out this work.

REFERENCES

[1] Cisco Systems, Reusable information object strategy, 1999.
[2] Gerard, R.W. (1967), "Shaping the mind: Computers in education", In N. A. Sciences, Applied Science and Technological Progress
[3] Polsani, P. (2003), "Use and abuse of reusable learning objects",
[4] Beck, Robert J., "What Are Learning Objects?", Learning Objects, Center for International Education, University of Wisconsin-Milwaukee, retrieved 2008-04-29.
[5] IBM Aglets, http://www.trl.ibm.com/aglets, Last visited: September 2005.
[6] IKV++Grasshopper,http://cordis.europa.eu/infowin/acts/analysys/products/thematic/agents/c h4/ch4.htm, Last visited: April 2006.
[7] Objectspace Voyager, http://www.recursionsw.com, Last visited: September 2005.
[8] Nerran, K, "Security in Mobile Agent Systems", http://www.cs.umn.edu/Ajanta, Last visited: March 2006.