Irish Journal of Academic Practice

Volume 3 | Issue 1 Article 9

2014-6

Speak Clearly, If You Speak at All; Carve Every Word Before You Let It Fall: Problems of Ambiguous Terminology in eLearning System Development

Damian Gordon
Technological University Dublin, Damian.Gordon@tudublin.ie

Deirdre Lawless
Technological University Dublin, deirdre.lawless@tudublin.ie

Claire Gordon
Technological University Dublin, claire.gordon@tudublin.ie

Recommended Citation

Gordon, Damian; Lawless, Deirdre; and Gordon, Claire (2014) "Speak Clearly, If You Speak at All; Carve Every Word Before You Let It Fall: Problems of Ambiguous Terminology in eLearning System Development," *Irish Journal of Academic Practice*: Vol. 3: Iss. 1, Article 9.
doi:10.21427/D7S42G
Available at: https://arrow.tudublin.ie/ijap/vol3/iss1/9

Speak clearly, if you speak at all; carve every word before you let it fall: Problems of Ambiguous Terminology in eLearning System Development

Damian Gordon, Deirdre Lawless, Claire Gordon

School of Computing
Dublin Institute of Technology

Abstract

This paper addresses issues associated with the development of eLearning software systems. The development of software systems in general is a highly complex process, and a number of methodologies and models have been developed to help address some of these complexities. Generally the first stage in most development processes is the gathering of requirements which involves elicitation from end-users. This process is made more complex by problems associated with ambiguous terminology. Types of ambiguous terminology include homonymous, polysemous and inaccurate terms. This range of ambiguous terminology can cause significant misunderstandings in the requirements gathering process, which in turn can lead to software systems that do not meet the requirements of the end-users. This research seeks to explore some of the more common terms that can be ambiguously interpreted in the development of eLearning systems, and suggests software engineering approaches to help alleviate the potentially erroneous outcomes of these ambiguities.

Keywords: Agile methodologies, Computer Science, Educational research, eLearning, Formal ontologies, Software Engineering, Terminology

Introduction

The paper is as result of an incident that occurred in 2004 when the lead author was working on a European project that focused on the development of an eLearning system to facilitate remote activation of scientific experiments. During a meeting between the Computer Scientists developing the system, and the Educational Researchers designing the system, a disagreement broke out over the prioritization of the metadata; the Computer Scientists insisted that the metadata was top priority, and nothing else could start until the metadata was agreed upon, whereas the Educational Researchers felt that the metadata was a relatively low priority concern, and there were other more vital considerations to prioritize first. This disagreement continued for over an hour until lead author of this paper joined the meeting, and asked for clarification as to the nature of the disagreement. Once explained he was quickly able to halt the disagreement by explaining to the group that the Computer Scientists and the Educational Researchers were discussing two very distinct topics that used the same term.

The challenges of managing the diversity and ambiguity of terminology in eLearning - a discipline at the nexus of Computer Science and Educational Research - has been explored by Anohina (2005), who concluded that the terminology is so widely used and ambiguous that it is vital to categorize and define relationships between the groups of terms to avoid confusion and design failures. Michaelson (2006) agrees with this assessment, and presents the failure of the UK e-university, known as UKeU, as a case study for evaluating the effectiveness of ICT, and cites misunderstandings of terminology, such as “Open Systems” (which was thought to be synonymous with “Open Source”) as a contributing factor to problems in the design process. Al-Ajlan & Zedan (2008) discuss the factors to consider when choosing a Virtual Learning Environment and acknowledge the terminological ambiguity will always be

https://arrow.tudublin.ie/ijap/vol3/iss1/9
DOI: 10.21427/D7S42G

an issue for "the world at large", but it is important that the relevant experts (Computer Scientists and Educational Researchers) have an agreed understanding of the issues.

This confusion is emblematic of the challenges that are encountered when any two disciplines interact, but also the challenges that become evident when a discipline that requires an exactness of specification (Computer Science) interacts with one that can tolerate more ambiguity (Educational Research). To understand these issues in a little more detail this paper will first present a little history of the key systems development methodology in Computer Science, *Software Engineering*, and then it will explore the history of eLearning to provide some contrast. Then the paper will explore examples of different kinds of ambiguous terms, and will finally suggest ways that these challenges can be overcome using structured approaches.

Software Engineering

Software engineering, the disciplined approach to the design, development and maintenance of software, has its roots in the 1950s with an early example of a development methodology being the Semi-Automated Ground Environment (SAGE) model in 1954 for the U.S. and Canadian air defense. Each decade subsequent to the 1950s resulted in a new wave of models and methodologies of Software Engineering yet it is interesting to note that it was not until the mid-late 1980s and early 1990s that these methodologies started to identify the end-user as being crucial to the development process (Boehm, 2006). Therefore for at least thirty years the needs of the end-user of the system were not considered pre-eminent in the process, but instead the development of software that could be created and modified easily by other programmers was considered paramount. This astonishing discrepancy has been addressed successfully since then by the assortment of techniques that have been developed since the

1990s that attempt to capture all of the end-user requirements by continually reverting to those end-users.

An important dimension to this process is *Requirements Engineering* which is concerned with identifying stakeholders relevant to the development process and identifying their needs. The elicitation process is so named as opposed to a term such as "requirements capture" since that would tend to suggest those requirements are out in the open and can be captured simply by asking the right questions, which would be misleading, whereas elicitation makes it clear that the requirements need to be drawn out (Nuseibeh & Easterbrook, 2000). There are a number of techniques to facilitate the elicitation process, which range from the *traditional techniques* of interviews, surveys and analysis of documentation, to *cognitive techniques* which are generally used for knowledge-based systems elicitation, to *contextual techniques* derived from qualitative techniques such as ethnomethodology and conversation analysis (Goguen & Linde, 1997). These techniques all rely on communications between human beings which by their very nature can result in ambiguity. There are certain domains of interest which can enhance the possibility for ambiguity, one of which is eLearning which will be introduced in the next section.

eLearning

eLearning is a relatively recent addition to the broad set of teaching approaches that have been developed since the first schools were founded over a thousand years ago. The first real instance of eLearning can probably be dated to 1924 when American psychology professor Sidney Pressey developed his "Testing Machine" (sometimes called the "Teaching Machine") which presented students with multi-choice questions which allowed them to choose their answer by pressing the appropriate button and this would be recorded on a sheet

https://arrow.tudublin.ie/ijap/vol3/iss1/9
DOI: 10.21427/D7S42G

of paper stored within the machine. Following this in 1931, American psychologist Burrhus Skinner developed an Operant Conditioning Chamber (or so-called "Skinner box") to teach animals to behave in specific ways, based on the principles of "programmed instruction" (Holmes & Gardner, 2006).

When computers became available to academic institutes in the early 1960s a movement amongst educators began to employ them not only from record-keeping but also for teaching. One of the earliest proponents of this approach American education philosopher Patrick Suppes who argued that computers could provide the one-to-one tuition that Benjamin Bloom demonstrated could improve student attainment by two standard deviations (i.e. moving a student from achieving 50% to 98%). Suppes founded the Computer Curriculum Corporation in 1967 which developed computer systems to teach elementary mathematics (Fernández Manjon *et al.*, 2006).

With the development of personal computers in the 1980s and the World-Wide Web in the 1990s, it became possible for educational institutes to fully harness the power of eLearning. During the 1980s single modules were first delivered online and then entire programmes were online, by the 1990s Virtual Learning Environments were being developed to provide tools to aid teachers in the development and management of their courseware (Weiss *et al.*, 2006).

It is significant to note that the majority of advances in eLearning have been accomplished by psychologists and educationalists as opposed to computer scientists, and as a consequence of this there has been a misappropriation of computer terminology by educationists and in return computer scientists have mistakenly used educational terms in their work. In the following

section some examples will be discussed in detail to identify potential pitfalls when educationists and computer scientists cooperate on the development of eLearning systems as a consequence of the cross-fertilisation of terminology.

Figure 1 The Pressey Testing Machine [source: Wikipedia]

Examples of Ambiguous Terminology

To ensure that educationalists and technologists can successfully co-operate in the development of eLearning materials, it is vitally important that there is a clear and unambiguous understanding and usage of terminology being used by both parties. Unfortunately it is the case that there are number of terms common to both fields that are used in different ways. This section will identify some of the ambiguous terms that have different meanings depending on context.

This ambiguity can be as a result of several scenarios, in the examples below three types of ambiguous terminology will be explored - homonymous, polysemous and inaccurate terms. Homonyms are words that have more than one distinct meaning. An even more complex problem can be caused by polysemes, which are words that have more than one meaning, but these meanings overlap somewhat. Inaccurate terminology is relatively common in eLearning

https://arrow.tudublin.ie/ijap/vol3/iss1/9
DOI: 10.21427/D7S42G

systems, where terminology is employed both from the education and computing fields, freely and often imprecisely.

Homonymous Term: "Learning Object"

A range of definitions exist as to what exactly a learning object is, the Learning Technology Standards Committee (LTSC, 2000) suggests that a learning object is "any entity digital or non-digital, which can be used, re-used or referenced during technology supported learning", The Learning Objects Network (LON, 2002) suggests that they are "stand-alone 'chunks' of information that have value. Examples include a text book; an appendix in another book; a map; a graphic; an interactive application; an online video; a wiring diagram; a simulation; and so on"; Such definitions are exceptionally broad and could almost be referring to anything. Wiley (2002, p.6) suggests a more useful definition of "any digital resource that can be reused to support learning."

The use of the term "object" is a cause of significant confusion as a result of the above definitions, in computing an *object* has a very specific meaning deriving from a family of programming languages classed as object-orientated. Object-orientated programming has been in existence for over 40 years, with the first true object-orientated programming language being *Simula 67* standardised in 1968 (Micallef, 1988). For the computer scientist an *object* is an instantiation of a class which is the blueprints or abstract characteristics of an entity, for example, for the class DOG an object could be LASSIE.

The object typically has a number of characteristics, including *Message Passing* which means object can pass information to other objects via messages, *Encapsulation* which means that the user of an object does not need to know everything about it and will only be exposed the

functions that they need, and *Polymorphism* which means that objects of different types have the ability to decide which of them need to respond to a message based on the message's context (Schach, 2006).

There is a clear difference between the more general definition of an object with can include anything from a simple diagram to an interactive application, and the more specifically defined concept of an object as understood from the computing perspective. This difference is ameliorated somewhat by the development of XML learning object standards (e.g. IMS, IEEE-LOM, Dublin Core) that can wrap both simple and complex learning materials, and enhance their functionality to incorporate some of the object-orientated features (Mohan & Daniel, 2004). Nonetheless there remains significant ambiguity in the usage and interpretation of the term "learning object."

Polysemous Term: "Metadata"

The term 'metadata' means "data about the meaning, content, organization, or purpose of data" (Siegel & Madnick, 1991, p.3). and generally in the educational context it is used to *"help a user understand what a* [resource] is about and can help a user decide if a specific [resource] is more or less likely to be relevant to his or her information need" (Cole & Foulonneau, 2007, p.111), or more simply put it is the background information associated with a learning resource. An example of metadata is a map legend, which can contain information such as the type of map, spatial references, the map's scale and its accuracy, the map publisher, the publication date, etc.

https://arrow.tudublin.ie/ijap/vol3/iss1/9
DOI: 10.21427/D7S42G

Figure 2 Map of Ireland, with enlarged map legend [source: paddysday.us]

In contrast when the term ‘metadata’ as being employed in the computing context it typically refers to how the data is constituted and how it relates to other data, and is frequently used in the context of *database schema* (Nijssen, Halpin, & Nijissem, 1989). Kashyap & Sheth (1998, p.164) state that “the function of the metadata descriptions is to be able to abstract out and capture the essential information in the underlying data independent of representational details.” So typically the metadata of a computer system will indicate whether the data is of numeric type or alphanumeric, and if it is numeric, is it stored as a natural number or a real number, i.e. will the number be recorded correct to a number of decimal places or not?

So, for example, if the data under consideration is a bank balance, it will have to be recorded to at least two decimal places, whereas if the data being recorded in the database was the total number of people who have deposited money into the bank in the a given month, this will be a whole number with no decimal places. In combination with this level of description, it is often necessary to describe the relationships between different data items - for example, a customer has a bank balance and a credit limit, and their balance should not exceed their limit.

This type of relationship can be expressed in many ways, some visual and some textual. Examples of visual techniques to show relationships between different data items include Entity Relationship Diagrams (Chen, 1976), Warnier/Orr diagrams (Warnier, 1981), CODASYL (Olle, 1978), and Class Diagrams (Fowler & Scott, 2000), examples of text-based techniques to describe relationships between data includes ASN.1 (Neufeld & Vuong, 1992), SQL (Chamberlin & Boyce, 1974), CBCL (McCarthy, 1982) and XML (Bray, Paoli, & Sperberg-McQueen, 1998).

XML, the Extensible Markup Language, has emerged as the *de facto* standard for describing and representing data on the World-Wide Web (Shanmugasundaram *et al.*, 1999). The principle reason for this is its versatility, unlike HTML which was its direct predecessor, XML allows computer scientists to create their own tags to describe the data. Additionally a number of complimentary specifications have been developed that support and extend the capabilities of XML, for example, DTDs (Document Type Definitions) describe the relationships between distinct XML documents, and XSL (the Extensible Stylesheet Language) which describes how to format and transform XML documents.

```
<soml version=1.1>
  <response type="Success" runid="1234">
    <param name="id" value="0001"/>
    <param name="alttext" value="Structure Included"/>
    <piggyback type="GetStructure">
      <param name="diaplay" value="TRUE"/>
      <arguments request="NewRun">
```


https://arrow.tudublin.ie/ijap/vol3/iss1/9
DOI: 10.21427/D7S42G

```
            <argument name="size" type="integer"
                        default="100"/>
        </arguments>
      </piggyback>
   </response>
</soml>
```

Figure 3 Example of Typical XML Metadata

Inaccurate Term: "Pre-internet"

Although the terms are frequently used synonymously, the "Internet" and the "World-Wide Web" are in fact two distinct technologies created over twenty years apart from each other. The origins of the Internet can be traced back to 1966, when MIT researcher Lawrence Roberts contacted the American Department of Defence *Advanced Research Projects Agency* (ARPA) to help fund the development of a computer network that would allow computers in various geographical locations to communicate, and to send and receive data. This led, in 1969, to the development of ARPANET, a proto-Internet, which initially linked four computers together.

By 1971, the network had grown to approximately 30 computers and the first e-mail was sent by researcher Ray Tomlinson, using the '@' symbol for the first time to indicate the address of a computer in the network. By 1973 specifications had been developed to allow for the transfer of files and voice traffic over this growing network. And in 1973-74 the TCP/IP protocols were developed which allowed for a more reliable of transfer of data across the network, and this led to of the Internet as we know it (Leiner *et al*., 1997), meaning that the Internet is at least 35 years old.

Decades after the development of the Internet, the World-Wide Web was first proposed by Tim Berners-Lee while working for CERN in 1989 as a "web of nodes" with "hypertext documents." He had successfully developed the first Web Server, the first Web Browser, and the first Webpage by the end of 1990, all of these technologies working over the Internet. This initial Webpage was composed of a combination of plain text with special tags interspersed to allow the browser to format the text. These tags are written in HTML (HyperText Markup Language) which describes the presentation of the text and allows the webpage author to specify hyperlinks in the text, and to include images on the webpage (Knuckles & Yuen, 2005).

Unfortunately this misunderstanding over terms has led to a number of educational researchers to incorrectly employ the term “Pre-internet” and undermine the accuracy of their research, and cause confusion with computer scientists. Some examples include:

- Mann (2000) suggests that there were two phases of education - “pre-Internet” and “Web-Based instruction” - leaving an unclassified hiatus of almost twenty years from 1973 (pre-internet) to 1991 (the first possible date of web-based instruction).
- Sellen (2002) confusingly refers to the “pre-Internet web world” an unfortunately incongruous combination.
- Enochsson (2005) erroneously claims that a study published in 1994 should be considered as “pre-internet” when in actuality the year 1994 is neither pre-Internet nor pre-Web.
- Mueller *et al.* (2006) suggests that journals published in 1992 should be considered as “pre-internet” which is incorrect by over 20 years.
- Kawachi, *et al.* (2006) claims that 1984 is the “pre-internet” era, incorrect by 10 years.
- Metros (2008) identifies 1991 as being in the “pre-internet” era, incorrect by 20 years.

https://arrow.tudublin.ie/ijap/vol3/iss1/9
DOI: 10.21427/D7S42G

- Sloep (2008) suggests that the early eighties were “still pre-Internet times” incorrectly.

More Examples of Ambiguous Terms

In this section we will briefly mention further terms that can be misinterpreted depending on the context in which they are being used, particularly when these terms are being used as labels in schedules or design specifications.

- **Active Learning:** In education this refers to a collection of techniques that focuses the responsibility of the learning on the student. In computers the term refers to a machine learning technique for classifying relationships between data.
- **CAT-6:** In education the term “CAT-6” is the California Achievement Test, a standardised test of basic skills used in the United States to test grades 2 through 11. In computing the term is used to refer to category 6 cable, a cable standard for Ethernet and other network protocols.
- **Cluster:** In education the term “cluster” refers to placing small groups of students together for instruction, in computers the term is used to mean a group of linked computers, working together to form a single computer.
- **Credentials:** In education the term “credentials” refer to qualifications such as diplomas, certificates and degrees, which attest to the successful completion of a specific educational program, whereas in computing the term is used to refer to a means of verifying identity to gain access to specific computer resources, such as passwords, fingerprints, and Public key certification.
- **eLearning:** The term “eLearning” itself is a somewhat ambiguous term, generally speaking computer scientists use the term to specifically refer to online mediated instruction, whereas in education the term is used to mean a much wider range of

approaches from lo-tech to hi-tech and from behavioristic to constructivistic. This term is explored in some detail by Romiszowski (2004).

- **Ontology:** in education the term "ontology" is used to describe a researcher's view of reality, in this sense it is often used with the term "epistemology", where the former is taken to mean "What is reality?" and the latter means "How can I measure reality?". In contrast, in computing the term is used to describe a formal definition of a hierarchy of concepts in a specific domain, using a shared vocabulary to denote the types, properties and interrelationships of those concepts.
- **Program(me):** In education the term "program" (or "programme") is used to refer to a program of study, e.g. one leading to a degree, based on a sequence of modules. In computing the term refers to a collection of instructions that a computer executes to perform a function or collection of functions.
- **Tracking:** In education the term "tracking" is used to describe the process of streaming students into different groups within an educational institute on the basis of their academic ability, whereas in computing the term is used in image processing to describe the process by which a corresponding point is located in a collection of images and this point of correspondence is interpolated in 3D space between these points.

Approaches to Addressing this Issue

Based on the issues discussed above it is clear that there is a rich of terminology that is shared between computing and education which mean different things in both domains therefore the traditional approaches to requirements gathering (e.g. interviews, surveys and analysis of documentation) can lead to ambiguity and therefore errors in design resulting in a developed eLearning system that does not achieve the requirements of the end-users. Therefore it is necessary to identify a set of approaches that will clarify this ambiguity.

https://arrow.tudublin.ie/ijap/vol3/iss1/9
DOI: 10.21427/D7S42G

Families of software development methodologies that are very much suited to addressing these types of issues are the Agile Family of Methodologies (Highsmith & Cockburn, 2001). The Agile methods are a collection of software development methodologies which focus on delivering working software over the processes by which this is achieved. The Agile Manifesto was drafted in February 2001, and identified the following four principles as vital:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

In terms of requirements gathering, one of the key ideas in all of the agile methodologies is the understanding that as well as identifying the key processes that are required, the context is king, so it is vital to understand the environments in which these processes occur, and how they impact each other and the end-users. This clearly would be beneficial in the eLearning domain. The most prominent technique used for capturing this context is *scenarios* (or "*scenarios of use*") which are essentially narratives that describe the user, their motivations, and their interactions with the system. The term 'scenario' in the computing context is a fairly generic term and scenarios can range from highly-detailed, e.g. ethnographic studies, to scant summaries, e.g. brief chats with users (Nardi, 1995).

One particularly popular scenario collection technique is User Stories, which consist of getting the user to describe a series of hypothetical stories in which they interact with the proposed system, and describe the functionality and interaction of the system. The stories are extremely powerful since they can provide clear, concrete examples of how the user wishes to interact with the system, and they provide nuance and details (Imaz & Benyon, 1999). This

approach to requirements gathering and design would be particularly apposite to eLearning development as the stories would help resolve many of the ambiguities associated with the collision between education and computing.

An example of this approach was used in the implementation of an eLearning system was being developed in Al-Ahliyya Amman University (AAU) in Jordan (Al-Yaseen, Hourani, & Al-Jaghoub, 2012, pp.58-64). When a number of challenges arose in the development process of the system, a move towards an agile approach was warranted. A coordinator on the project stated: "An e-learning project is a special case and it is different from any other IS project because of the involvement of the human factor and the differences between disciplines".

An interesting alternative which can be considered as antithetical to User Stories in the context of formality of approach, are ontologies. Ontologies are formal descriptions of concepts and the interrelationships between them, in a specific universe of discourse. These ontologies can be expressed in a wide range of ways, but typically they are expressed in a particular dialect of XML (such as in Figure 3 above). The most significant benefit to this approach is that it provides computer-readable data that can be subjected to automated validation. Additionally by collecting requirements from a range of end-users and encoding them as ontologies, it is possible to merge the ontologies to create a larger overall ontology which again can be used for cross-validation purposes.

If two users were referring to "metadata" and it was clear from the descriptions that the two users are referring to two distinct conceptualisations of metadata, it would be possible to classify 'metadata' as two distinct concepts METADATA[1] and METADATA[2] and each would be distinguished under different semantic classes (Happel & Seedorf, 2006). This

https://arrow.tudublin.ie/ijap/vol3/iss1/9
DOI: 10.21427/D7S42G

approach to requirements gathering would be a viable alternative to User Stories, and would address the issues of ambiguity in eLearning terminology.

Cakula & Salem (2013, pp.14-25) present the application of the ontological engineering methodology in the development of two web-based eLearning ontologies to teach the topic of artificial intelligence. They warn that the educational researcher must "work as a knowledge engineer making the skeleton of the studied discipline visible and showing the domain's conceptual structure", again highlighting the need for computer scientists and educational researchers to work together to develop an agreed understanding of the issues under consideration.

These two approaches to the requirements gathering process, *User Stories* and *Ontologies*, are representative examples of methodologies that can help disambiguate user requirements; there are many other approaches possible, e.g. triadic sorting, mindmaps, laddering, but whichever approach is chosen it is vital that the methodology eliminate as much ambiguity as possible.

Conclusion

In this paper a brief history of eLearning was outlined to illustrate the fact the eLearning is a collaboration of work by computer scientists and educationists. This collaboration from two distinct disciplines can potentially result in ambiguities as a result of terminologies being used differently by these practitioners and this in turn will lead to an eLearning system being developed that does not address all of the requirements of the users. This paper proposes that an appropriate way to address this situation is at the beginning of the development process, by ensuring that the requirements gathering process is done correctly many problems will be

circumvented, thus for eLearning systems the appropriate requirements gathering process needs to be highly discriminating in terms of understanding terminology and its underlying concepts.

https://arrow.tudublin.ie/ijap/vol3/iss1/9
DOI: 10.21427/D7S42G

References

Al-Ajlan, A., & Zedan. H. (2008). Why Moodle. *Future Trends of Distributed Computing Systems*. FTDCS'08. 12th IEEE International Workshop on IEEE. 21-23 October, 2008.

Al-Yaseen, H., Hourani, M., & Al-Jaghoub, S. (2012). Success and Failure of e-Learning Projects: Alignment of Vision and Reality, Change and Culture, *Journal of Emerging Trends in Computing and Information Sciences*, 3(2), 277-284.

Anohina, A. (2005). Analysis of the terminology used in the field of virtual learning. *Educational Technology & Society*, 8(3), 91-102.

Boehm, B. (2006). A View of 20th and 21st Century Software Engineering. *Proceedings of the 28th International Conference on Software Engineering*, 20-28 May, 2006.

Bray, T., Paoli, J., & Sperberg-McQueen, C.M. (1998). *Extensible Markup Language (XML) 1.0*. Retrieved: 11, December 2013 from http://www.w3.org/TR/REC-xml

Cakula, S., & Salem, A.-B.M. (2013). E-Learning Developing Using Ontological Engineering. *WSEAS Transactions on Information Science and Applications*, 10(1), 14-25.

Chamberlin, D.D., & Boyce, R.F. (1974). SEQUEL: A Structured English Query Language. *Proceedings of the 1974 ACM SIGFIDET Workshop on Data Description, Access and Control* (pp.249-264).

Chen, P. (1976). The Entity-Relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1(1) 9-36.

Cole, T.W., & Foulonneau, M. (2007). *Using the Open Archives Initiative Protocol for Metadata Harvesting*. Connecticut, USA: Libraries Unlimited Inc.

Enochsson, A. (2005).The Development of Children's Web Searching Skills - A Non-Linear Model. *Information Research*, 11(1), paper 240.

Fernández Manjon, B., Sanchez Perez, J.M., Gómez Pulido, J.A., Vega Rodriguez, M.A., Bravo, J. (Eds.), (2006). *Computers and Education: E-learning - From Theory to Practice*. Germany: Kluwer.

Fowler, M., & Scott, K. (2000). *UML Distilled.* Reading: Addison-Wesley.

Goguen, J.A., & Linde, C. (1997). Techniques for Requirements Elicitation, *Software Requirements Engineering* (pp.110-122), Second Edition. Washington, DC: IEEE CS Press.

Happel, H-J., & Seedorf, S. (2006). Applications of Ontologies in Software Engineering. In *2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006),* Athens, GA, USA, 12-14 November, 2006.

Highsmith, J., & Cockburn, A. (2001). Agile Software Development: The Business of Innovation. *IEEE Software,* 34(9), 120-122.

Holmes, B., & Gardner, J. (2006). *e-learning: Concepts and Practice*. London: Sage Publications.

Imaz, M., & Benyon, D. (1999). How Stories Capture Interactions, in *Proceedings of Human-Computer Interaction - INTERACT'99* (pp.321-328). Netherlands: IOS Press.

Kashyap, V., & Sheth, A. (1998). Semantic heterogeneity in global information systems: the role of metadata, context and ontologies. In M. Papazoglou & G. Schlageter (Eds.), *Cooperative Information Systems: Current Trends and Directions* (pp.139-178). United Kingdom: Academic Press.

Kawachi, P., Sharma, R.C., & Mishra, S. (2006). The Asian Age Variable in Open and Distance Education. *Asian Journal of Distance Education*, 4(1), 1-3.

Knuckles, C.D., & Yuen, D.S. (2005). *Web Applications: Concepts and Real World Design.* New York, USA: John Wiley and Sons.

Leiner, B., Cerf, V., Clark, D., Kahn, R., Kleinrock, L., Lynch, D., Postel, J., Roberts, L., & Wolff, S. (1997). The Past and Future History of the Internet. *Communications of the ACM*, 40(2), 102-108.

LON (2002). *Learning Objects Network Site.* Retrieved: 11, December 2013 from http://www.learningobjectsnetwork.com

LTSC (2000). *Learning Technology Standards Committee Site*. Retrieved: 11, December 2013 from http://ltsc.ieee.org

Mann, B.L. (2000). Phasing-In To Online Global Education: A Key To Successful Web Course Management. In *Proceedings of SSGRR 2000, International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, L'Aquila, Italy, 31st July-6th August, 2000.

McCarthy, J. (1982). Common Business Communication Language. In A. Reetz & J.R. Oldenbourg (Eds.), *Textverar- beitung und Burosysteme, Endres*. Munich and Vienna: Verlag.

Metros, S.E., (2008). The Educator's Role in Preparing Visually Literate Learners. *Theory into Practice*, 47(2), 102-109.

Micallef, J. (1988). Encapsulation, Reusability, and Extensibility in Object-Oriented Programming Languages. *Journal of Object-Oriented Programming*, 1(1), 12-36.

Michaelson, R. (2006). Evaluating Failure: the case of UKeU. *Proceedings of the 7th Annual Conference of the Higher Education Academy for Information and Computer Sciences*, Trinity College, Dublin, 29th- 31st August, 2006.

https://arrow.tudublin.ie/ijap/vol3/iss1/9
DOI: 10.21427/D7S42G

20

Mohan, P., & Daniel, B. (2004). The Learning Objects' Approach: Challenges and Opportunities. In G. Richards (Ed.), *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (pp. 2512-2520), AACE.

Mueller, P.S., Murali, N.S., Cha, S.S., Erwin, P.J., & Ghosh, A.K. (2006). The Effect of Online Status on the Impact Factors of General Internal Medicine Journals. *Netherlands Journal of Medicine*, 64(2), 39-44.

Nardi, B. (1995). Some Reflections on Scenarios. In J.M. Carroll (1995). *Scenario-based Design*. New York, USA: John Wiley.

Nijssen, G.M., Halpin, T.A., & Nijissem, S. (1989). *Conceptual Schema and Relational Database Design*. London: Pearson Education.

Neufeld, G.W., & Vuong, S. (1992). Overview of ASN.1. *Computer Networks and ISDN Systems*, 23(5) 393-415.

Nuseibeh, B., & Easterbrook, S. (2000). Requirements Engineering: A Roadmap. In A.C. Finkelstein (Ed.), *The Future of Software Engineering (the proceedings of the 22nd International Conference on Software Engineering, ICSE´00)*. IEEE Computer Society Press.

Olle, T.W. (1978). *The CODASYL Approach to Data Base Management*. New York, USA: Wiley.

Romiszowski A.J. (2004). How's the E-learning Baby? Factors Leading to Success or Failure of an Education Technology Innovation. *Education Technology*, Jan-Feb, 5-48.

Schach, S. (2006). *Object-Oriented and Classical Software Engineering.* New York, USA: McGraw-Hill.

Sellen, M.K. (2002). Information Literacy in the General Education: A New Requirement for the 21st Century. *Journal of General Education*, 51(2), 115-126.

Shanmugasundaram, J., Tufte, K., He, G., Zhang, C., DeWitt, D., & Naughton, J. (1999). Relational Databases for Querying XML Documents: Limitations and Opportunities, *Proceedings of the 25th International Conf. on Very Large Data Bases* (pp.302-314). New York, USA: Very Large Data Bases Endowment Inc.

Siegel, M., & Madnick, S. (1991). A Metadata Approach to Resolving Semantic Conflicts, *Proceedings of 17th International Conf. on Very Large Data Bases* (pp.133-145). New York, USA: Very Large Data Bases Endowment Inc.

Sloep, P.B. (2008). Preface to the Special Issue: Comparing Educational Modelling Languages on the 'Planet Game' Case Study. *Journal of Interactive Media in Education*, 1(10), pp.vii-x.

Warnier, J-D. (1981). *Logical Construction of Systems.* London: Thomson Learning.

Weiss, J., Nolan, J., Hunsinger, J., & Trifonas, P. (Eds.), (2006). International Handbook of Virtual Learning Environments. Washington, USA: *Springer International Handbooks of Education*, 14.

Wiley, D.A., (Ed.), (2002). *The Instructional Use of Learning Objects*. Indiana, USA: Agency for Instructional Technology.

https://arrow.tudublin.ie/ijap/vol3/iss1/9
DOI: 10.21427/D7S42G