

Appended papers

Paper 1 Paulsson, F., & Naeve, A. (2007). Establishing technical quality criteria for Learning Objects. In P. Cunningham & M. Cunningham (Eds.), *Exploiting the Knowledge Economy: Issues, Applications, Case Studies* (Vol. 3, pp. 1431-1439). Amsterdam: IOS Press.

Paper 2 Paulsson, F., & Naeve, A. (2006). Virtual Workspace Environment (VWE): A Taxonomy and Service Oriented Architecture Framework for Modularized Virtual Learning Environments - Applying the Learning Object Concept to the VLE. *International Journal on E-Learning*, 5(1), 45-57.

Paper 3 Paulsson, F., & Berglund, M. (2008). Suggesting a SOA-framework for modular virtual learning environments: comparing two implementation approaches. *International Journal of Web-Based Learning and Teaching Technologies*, 3(1), 43-57.

Paper 4 Paulsson, F., & Berglund, M. (2006). A service oriented architecture-framework for modularized virtual learning environments. In A. Méndez-Vilas, A. Solano Martin, J. Mesa Gonzalez & J. A. Mesa Gonzalez (Eds.), *Current Developments in Technology-Assisted Education* (1 ed., Vol. 3, pp. 1935-1939): FORMATEX.

Paper 5 Paulsson, F., & Engman, J. (2007). Treating metadata as annotations: separating the content mark-up from the content. *International Journal of Emerging Technologies in Learning (iJET)*, 1(3), 1-7.

Paper 6 Paulsson, F., & Engman, J. (2005). Marking the National Curriculum - a new model for semantic mark-up. In P. Cunningham & M. Cunningham (Eds.), *Innovation and the Knowledge Economy: Issues, Applications and Case Studies* (Vol. 2, pp. 1731-1738). Amsterdam: IOS Press.

1 Introduction and background

The research area, together with the background of my research and the problems studied are presented in this chapter. Some basic concepts that are used in this thesis are introduced. The structure, the research questions, the purpose and delimitations of this thesis are also described.

1.1 Technology Enhanced Learning and learning technology

The use of technology to support learning has been around for quite some time now. The kind of technology, and the ways technology have been used has varied over time. Nowadays, when Technology Enhanced Learning (TEL) is discussed, it is usually learning with the help of ICT that is targeted.

The use of computers for learning and education started to emerge in the 1950's, emanating from efforts to train soldiers in a fast and efficient way during the Second World War (see, e.g., Saettler, 2004, pp. 178-196). Behaviorists like Skinner and Thorndike were in the lead of the early efforts that eventually led to the idea of instructional design. Instructional design consists of several related theories and concepts that are rooted in learning theories. Instructional design theory deals with the methods of instruction; while instructional models, instructional strategies, etc., represent the practice of applying the theory to instructional development or instructional development processes (Reigeluth, 1999c).

The instructional design movement continued to evolve during the later half of the 20th century and it has made deep impressions in the TEL history with ideas such as the Teaching Machine, first introduced by Skinner (1958), later re-launched by Gagné (1965). Other milestones were the concepts of *programmed instruction* that were intended to be realized by the Teaching Machine, based on Skinner's *frames of instruction*, described by Skinner (1958) and *programmed learning* that was an evolvement of programmed instructions. Later on, during the sixties, seventies, and eighties, programmed learning evolved into computer-based training and computer-based learning, and during the seventies learning theory and pedagogical practice started to shift from behavioristic theories to cognitivists' learning theories (see, e.g., Gagné, 1977).

One of the most significant technological developments during the eighties was the introduction of affordable desktop computers that opened up for the use of technology in most organizations – including schools. Desktop computers made it possible to introduce pedagogical software at the classroom level.

The development and use of TEL exploded during the nineties and the breakthrough of multimedia and CD-ROM technology in the early nineties opened up new possibilities for the delivery of digital learning content. In parallel, the development of hypertext - that started in forties with the work of Bush (1945) and gained speed in the sixties, when Nelson (1965) coined the term hypertext – eventually led to the development of the World Wide Web (the web), (see, e.g., Nielsen, 1995), which revolutionized TEL. Suddenly TEL was a global affair, and it became possible to distribute collaborative learning and digital learning content in ways that were unmanageable just a decade before. The web shifted the focus from *Computer-Based Training (CBT)* to *Web-Based Training (WBT)* or *Web-Based Learning (WBL)* – depending on the pedagogical point of view. During the eighties and nineties, the pedagogical focus shifted once again, this time to the benefit of constructivism, influenced by the work of researchers like Vygotskij and Piaget (see, e.g., Mylov, 2000; Dusek, 2006, pp. 198-210).

The breakthroughs of the Internet and the web marked the beginning of a general globalization that changed the way ICT-systems and system borders are perceived. ICT-

systems and their functionality have become a global issue that goes beyond organizational boundaries. Consequently, system functionality becomes distributed and intertwined in a way that creates dependencies that are beyond the control of a single organization. This has triggered a development of new types of flexible systems and services, and has at the same time created a need for common standards and frameworks for open interfaces. Altogether, those developments have led to a modularization of ICT infrastructures and systems.

New ideas and strategies have been applied in order to meet the new demands for flexibility and adaptability, and this has triggered new ways of implementing systems. One of the most significant changes during the last decade is an increased level of system modularity. Modular web-based applications are often accomplished by exposing (providing) functionality as services to the outside world, such as Web Services in a web-based *Service Oriented Architecture (SOA)*. Services can subsequently be used (consumed) by other systems or services as loosely coupled functionality-components within a larger pool of services. The usage contexts are provided by the way in which services are being used, and a service may be used for different purposes in different contexts. This way it is possible to create flexible and dynamic application-like functionality on the web using modular approaches where services interrelate through the utilization of each other, creating a global grid of functionality and information.

Web Service-based SOA-approaches have mainly been applied to legacy systems, but are also slowly changing how new systems are designed and implemented. Modular infrastructures and system architectures are becoming more common in new development as well, and general reference models and modular frameworks are emerging within several application areas, including TEL, where these create the foundation for new kinds of distributed *Virtual Learning Environments (VLE)*. Examples of such SOA-based frameworks are the *e-Framework*, described by Wilson, Olivier, Jeyes, Powell, and Franklin (2004), and the *Open Knowledge Initiative's (OKI) Open Service Interface Definitions (OSIDs)* that are used in the Sakai project, described by Thorne and Kahn (2003).

In addition to Web Service-based approaches there are other more lightweight approaches to exposing or syndicating functionality and data that are already used in VLEs. Examples are technologies such as XML-RPC or RSS¹, which are used for simpler, but yet powerful, services in isolated contexts. Those technologies support modularity as well, as they can be used to syndicate a specific service or specific data to be used by other services or applications.

There is a comprehensive amount of work that implies that increased modularity is a path to a considerably higher level of flexibility, from the technological point of view, as well as from a usage point of view in terms of how functionality and information can be accessed, assembled and utilized in different contexts (see, e.g., Williams, 2001; Baldwin & Clark, 2000; Nierstrasz & Dami, 1995). Flexibility can also be contemplated from a pedagogical point of view, and pedagogical flexibility is something that is needed in order for technology to meet new and changing pedagogical requirements.

Modular approaches for TEL have mainly been applied to digital learning content, and *Learning Objects* is one of the most widespread and widely adopted concepts for delivery of such content. Learning Objects represent a modular approach that aims at facilitating interoperability, flexibility and reuse of digital learning content. The positive relation between modularity and flexibility can be derived from research on learning objects as well. Abernethy, Treu, Piegari, and Reichgelt (2005), uses the case of introductory ICT courses to

¹ See Appendix 1 for explanations of technical terms.

show how enhanced flexibility is achieved using modular learning content. They emphasize the relation between the level of granularity of learning objects and the achieved flexibility. Rehak and Mason (2003), point out the benefits of modularity, while stressing the problems caused by differences in how the learning object concept is perceived in the learning community, (see also, e.g., Gibbons & Nelson, 2002; Silveira et al., 2007). These citations are merely examples of work that emphasizes the relation between modularity and pedagogical flexibility. The dilemma is not to establish that there is such relation, the problem is rather that the views of what flexibility and adaptability are and how it can be achieved and utilized are ambiguous.

The emergence of the learning object concept has triggered a change in how VLEs are perceived and implemented. The idea of Learning Management Systems (LMS) is intimately connected to learning objects, and LMSes were originally a way of placing learning objects into a clearer and more suitable context for education and training in trade and industry. It is therefore common for LMSes to focus on training and performance rather than on learning and education. Many LMSes are more focused on providing functionality for administration and management of learning, courses, and learning objects, rather than on providing advanced features for pedagogical activities (see, e.g., Ismail, 2002). The pedagogical context is often provided as a part of the learning content governed by the way in which the content is sequenced and packaged. The result is that much of the application logic providing pedagogical support is coded inside digital learning content.

The term LMS is sometimes used synonymously with the term VLE, and this causes a lot of confusion. As an example, Paulsen (2002) defines a Learning Management system as:

“Learning Management System is a broad term used for a wide range of systems that organize and provide access to online learning services for students, teachers, and administrators. These services usually include access control, provision of learning content, communication tools, and organization of user groups.”

In this thesis LMSs are regarded as being subtype of VLEs. LMSs are further discussed in chapter 3.

In recent years, the focus has shifted toward social interaction, user-created content and different forms of user participation. The view of the learner is transforming from being a user of the VLE to being a participator in distributed VLE communities, combined with technological developments, changing attitudes, open services on the web, etc, a change that is sometimes referred to as *Web 2.0* (see, e.g., O'Reilly, 2006; Anderson, 2007). This evolvement toward social interaction and participation corresponds well to the shift of pedagogical focus toward constructivist learning theories that has occurred during the last 10-20 years (see, e.g., Jonassen, 1991c; Dempsey & Van Eck, 2007).

1.1.1 The future is semantic

Parallel to the development described in the previous chapter, the Semantic Web envisioned by Berners-Lee, Hendler, and Lassila (2001), is evolving. The Semantic Web is changing the ways in which information management and system interaction can be done, creating new exciting possibilities for knowledge management and TEL that are likely to completely change the way in which we regard information searching and browsing. One such example is the concept browser *Conzilla*, described by Naeve (2001). The semantic web will most definitely be essential for TEL and it will have implications for how Web Services and modular systems can be assembled and kept together (see, e.g., Nilsson, 2001; Nilsson, Naeve, Duval, Johnston, & Massart, 2008; Naeve, 2005; Naeve & Sicilia, 2006; Dietze, Gugliotta, & Domingue, 2007; Richter, Nejd, & Allert, 2002; Koper, 2004; Devedzic, 2004).

Semantic Web Technology is already used to create *Semantic Web Services*, i.e., self-describing services that can be automatically discovered and matched with other services and information for automatic web service composition without intervention of humans (see, e.g., Nagarajan, 2006a; Lama, Arroyo, Sicillia, & Lópes-Cobo, 2006; Dietze, Gugliotta, & Domingue, 2007).

The technical foundation for the Semantic Web is an information description language called *Resource Description Framework* (RDF) (Decker et al., 2000). RDF's abilities to represent data and information semantics make it possible to transfer some of the programmatic and logical capabilities of application software to be represented by information semantics using RDF, and be completely detached from application software. Instead it can be interpreted by RDF-aware software and incorporated in different services. One consequence of this is that software components and software services can act (and re-act) differently depending on the semantics expressed by RDF-data. These are especially interesting capabilities for dynamic and modular environments, and for representing dynamic and generative processes independently of software and application logics. Cardoso and Sheth (2006a), and Fensel and Bussler (2002) describe the concept of combining Semantic Web technology and Web Services into *Semantic Web Services*.

1.2 Emerging problems

The technical developments leading up to new ways for delivering TEL, described in chapter 1.1, together with a shift in pedagogical focus have led to a veritable revolution of learning and education. New possibilities have arisen, but also new challenges and problems.

A change in pedagogical focus has occurred parallel to the technological development. While Behaviorists' learning theories were predominant during the early to mid-20th century, there has gradually been a shift toward cognitivistic and constructivistic learning theories during the later half of the 20th century (see, e.g., Schunk, 2003). Constructivistic approaches and associated pedagogical methods, such as problem-based learning and case-based learning, are becoming increasingly popular among teachers and pedagogues in all parts of society. The correspondence between behavioristic learning theories and the learning technology used in computer-based training and other similar approaches is quite good, while the pedagogical shifts during the later half of the 20th century have contributed to the creation of a gap between the pedagogical requirements and expectations and the supporting technology.

Many pedagogues discern interesting possibilities in the use of technology, but the change in pedagogical focus and the variety of pedagogical requirements introduce new demands on how technology must be implemented, as well as on how pedagogical design must be realized and represented by technology. Much of the technology in use today still bears a legacy from its early behavioristic roots, focusing on instruction and training rather than on learning. The consequence is that a large portion of the learning technology in use does not match pedagogical expectations and requirements. Learning technology needs to become more flexible, both from a pedagogical point of view and from a technological point of view. Pedagogical flexibility is dependent on technological flexibility, which means that learning technology must be able to support and adapt to a variety of pedagogical approaches, as well as to different functional requirements provided by these approaches.

While an increased level of modularity provides a way of increasing flexibility, LMSes often fail to meet the requirements for flexibility through modularity. LMSes are often implemented in a silo-like fashion, unable to adapt to a modular approach. As information silos, they do not interact well with the surrounding world and this problem becomes worse as much of the surrounding world consists of other information silos. Those information silos often contain

information and functionality that would be useful for the LMSes, and this leads to a need to duplicate data and functionality, creating unwanted redundancy (see, e.g., Downes, 2002).

The situation becomes even more complicated as learning objects provide a modular concept for content, while there is no corresponding model or concept for the VLE, which means that the potential flexibility provided by modularity cannot be utilized to its full potential. Hence, modular concepts aiming for reusable and flexible digital learning content are being implemented in non-modular and inflexible VLEs. For many reasons this is a non-fit, and much of the pedagogical benefits that could be obtained through modularity are lost when using non-modular VLEs. Current practices with LMSes merely use the VLE as a container for unpacking and displaying digital learning content that is instructionally designed and sequenced beforehand, while missing out on many of the possibilities that are created by modular content in combination with modular functionality (see, e.g., Dodani, 2002; Abdallah, El Hajj, Benzekri, & Moukarzel, 2002; Downes, 2002; Edwards, Rai, Philips, & Fung, 2007).

Both within the fields of computer science and TEL, modularity is often ‘sold’ using arguments like reuse, flexibility and adaptability (see, e.g., Silveira et al., 2007; Polsani, 2003; Dodani, 2002). At the same time learning objects are criticized for not meeting the expectations of modularity, reusability, and platform and pedagogical neutrality. Learning objects simply do not deliver what they are expected to deliver (see, e.g., Dodani, 2002; Polsani, 2003; Downes, 2002).

In this thesis, I argue that there is a good and simple reason for that: the expected characteristics of learning objects are well expressed by the learning object community but they have failed to express how those characteristics should be implemented in order to be accomplished. This becomes even more evident when comparing learning objects to component technologies within computer science, where modularity is essential for flexibility in terms of reuse, dynamics and adaptability.

Ideas and concepts from computer science were the original source of inspiration for the learning object community when the concept was first invented and are often referred to when discussing learning objects and modularity, such as by Song and Andersson (1999) and Dodani (2002). The definitions of modularity and related concepts, as well as how they should be implemented are much clearer within computer science. There are clear rules, models and frameworks that make it all work. Szyperski (2002c) defines a software component (i.e., a type of module) as:

“... a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties.” (p. 36)

A problem that the learning object community is struggling with is that some of the most important characteristics that define components have been left unspecified. The result is that ideas are falling apart and concepts and definitions are becoming fuzzy and straggly. Those issues can be regarded as problems that concern the technical quality, and there are many lessons to be learned from computer science.

However, modularity is only one piece of the puzzle. In general learning is not a linear and homogenous process, which means that there is also a need for new ways of representing pedagogical processes in VLEs, processes that utilize the potential flexibility that is created by modularity. Current practice of sequencing, such as provided by SCORM *Sequencing and Navigation* (Capone, Christensen, Falls, & Fletcher, 2004) is not powerful enough. One of the main reasons for this is that sequencing assumes that the pedagogical sequence and process

can be pre-determined. Attempts have been made in order to develop more sophisticated models for representing pedagogical process. The most common approach has been through the use of *meta-languages*, so-called *educational modeling languages* (EML) (see, e.g., Rawlings, Koper, van Rosmalen, Rodrigues-Artacho, & Lefrere, 2002). One of the more recent attempts is the *IMS² Learning Design* (Koper, 2005). Even though educational modeling languages provide considerably more flexibility, they only solve a portion of the problem, as the pedagogical process still needs to be pre-determined to a large extent.

A conclusion that can be drawn from the reasoning above is that a high level of modularity provides a sound basis for achieving technical flexibility, and that technical flexibility can be utilized to achieve pedagogical flexibility. Furthermore, there is a need to establish an infrastructure that can provide the conditions needed to reach the same level of modularity for the virtual learning environment as learning objects are intended to provide for digital learning content.

My hypothesis is that a reasonable approach would be to incorporate digital learning content and virtual learning environments into the same conceptual space, using a common framework and a common abstract model. To utilize the potential flexibility that is provided by a modular infrastructure there is a need to represent pedagogical design and pedagogical process in ways that are generative rather than normative, which is the case with most current practice.

² IMS Global Learning Consortium, <http://www.imsproject.org/>

1.3 Research questions

Based on the problems outlined above, the following research questions have been identified:

1. *How can a common abstract model for modularity, which covers both the VLE and digital learning content, be accomplished without creating incompatibilities with existing models?*
2. *How can a technical infrastructure be designed and implemented according to a common abstract model, in order to provide the necessary supporting infrastructure, and without disqualifying legacy?*
3. *How can pedagogical designs and processes be represented in order to utilize modularity to increase pedagogical flexibility and adaptability?*

1.4 The purpose of this thesis

The overall aims of this thesis are:

To examine how, and to which extent, modularity can help in enhancing pedagogical and technical flexibility.

To show how experience from computer science and system architecture can be applied to learning objects and the VLE, by establishing links between modularity in computer science and modularity in pedagogical practice.

These aims will be accomplished using an experimental approach, based on the reasoning in chapter 1.2, by examining modularity through implementation in software systems and infrastructures.

1.5 Thesis structure

The first three chapters provide the background, context, and theoretical framing for the research presented in this thesis, while the last two chapters account for the results, conclusions and discussion of the work.

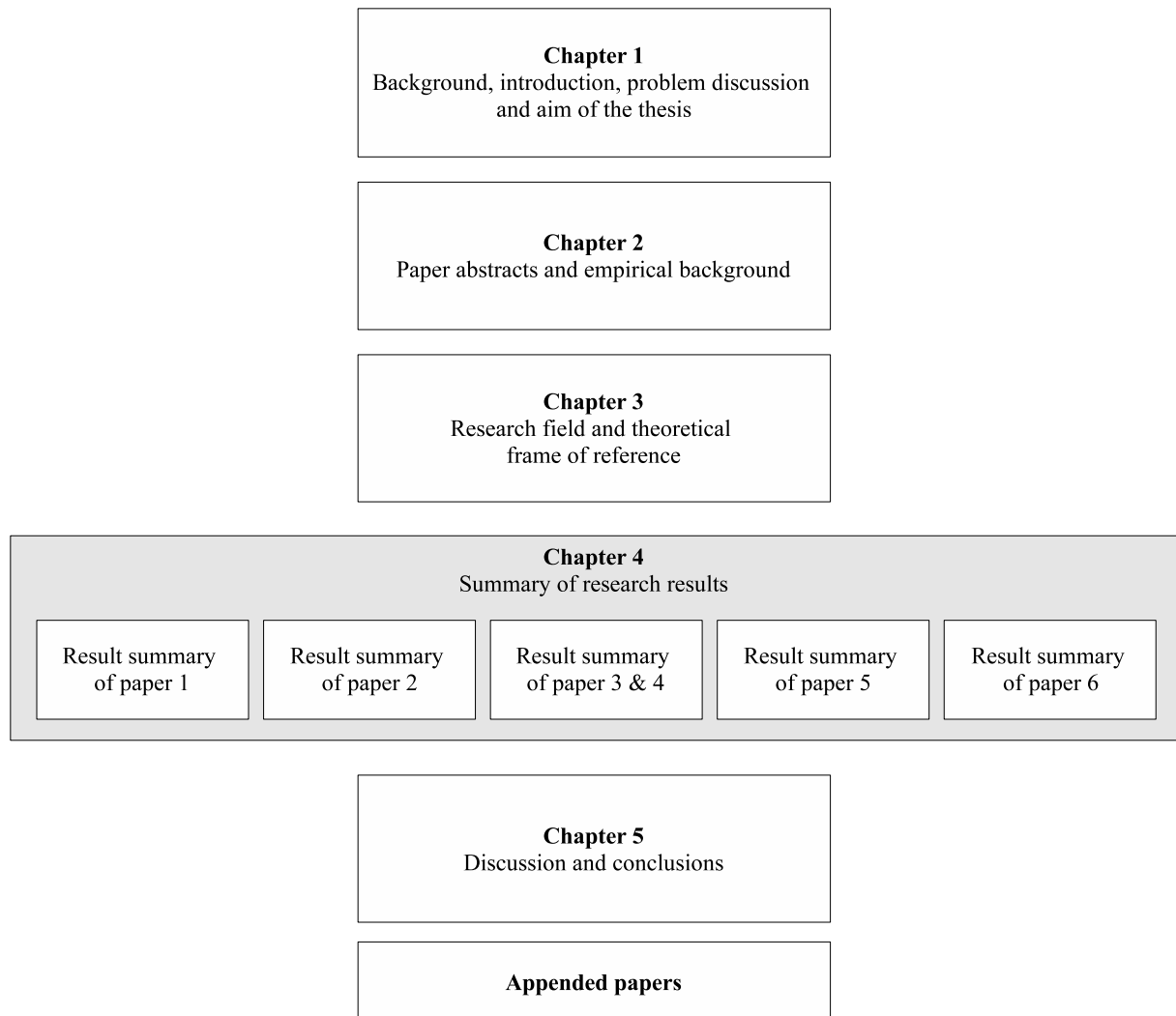


Figure 1-1. The structure of this thesis.

The first chapter gives an introduction to the research field and a description of the problem area. The research questions and the aim of the thesis are also presented in chapter 1. Chapter 2 summarizes the papers and gives an overview of the projects that constitutes a large part of the empirical context of this thesis. Chapter 3 contains an overview of the research field and provides the theoretical framing. The research results and my contributions to the field are presented in chapter 4, after which my conclusions and discussion of the results follow in chapter 5.

The structure of the thesis is shown in Figure 1-1 and the mapping of the research questions and the appended papers are shown in Figure 1-2.

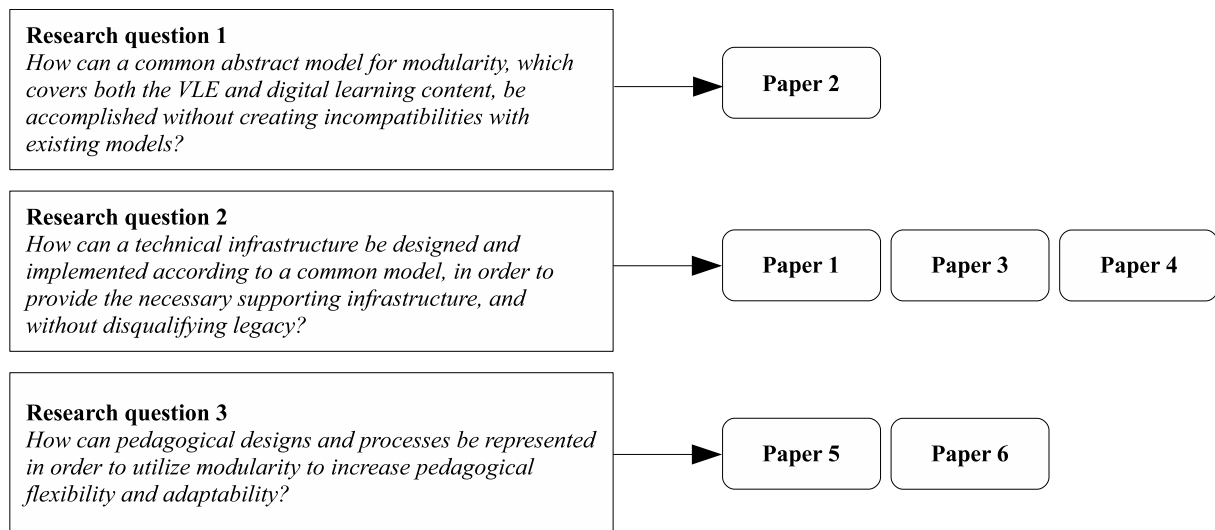


Figure 1-2. The three research questions and how they are addressed in the appended papers.

1.6 Delimitations

Even though this thesis emanates from a pedagogical point of view, includes pedagogical contexts and discusses pedagogical theories, the research has a strong focus on technical and technological consequences of pedagogical requirements. Hence, the pedagogical aspects are to be regarded as the empirical context of the technological research presented in this thesis. The main focus is on infrastructure, modularity and system architecture for virtual learning environments and digital learning content used for TEL.

The purpose is neither to push any pedagogical values nor to examine the implications of different pedagogical approaches. The only relevant aspect for the context of this thesis is that different pedagogical theories and approaches exist, and that this undisputable fact has consequences for the design and development of the learning technology to be examined.

Furthermore, this thesis does not compare general technical characteristics (such as performance) or detailed technical differences in the implementation strategies of technological frameworks, other than in cases where this is relevant in a TEL context or from a modularity point of view.

1.7 Concepts and terms

This thesis is interdisciplinary in many ways, and I reserve for myself the use of some terms and concepts in a slightly unorthodox way depending on the context. This has its explanation in the fact that some of the concepts and terms are used differently in the TEL community than within computer science. Some terms have a slightly different meaning also within computer science, depending on whether they are used in, for example, component-based software engineering, or in a web application context, such as with web services or in a Web 2.0 context.

The rest of this chapter is devoted to briefly explaining some basic terms that will be further discussed in the coming chapters, while other terms and concepts will be explained as they are introduced.

1.7.1 Learning technology

The term *learning technology* will be used for referring to the technology that is used for learning. The term learning technology puts the focus on *learning* that can be either formal or informal, and not on *education*, which can be interpreted as a more formal form of learning, as in the term *educational technology*, used by Januszewski and Yeaman (2001). The term learning technology can be interpreted as being fairly pedagogically neutral as it avoids the tension of terms like *instructional*, as in *instructional technology* that leads the thoughts to instructional approaches, such as *instructional design* or *individual instruction*. Seels and Richey (1994) use the term instructional technology.

Learning technology is a general term that covers all aspects of technology used for learning in all contexts, including technology that is mainly intended for a specific context or approach, such as approaches with an instructional focus. For the purpose of this thesis, the term learning technology only refers to digital technology, i.e., ICT, even though learning technology could be argued to embrace analog technology as well.

1.7.2 Technology Enhanced Learning

While the term learning technology is used to refer to the *technology* used for learning, the term *technology enhanced learning (TEL)* will be used to refer to the actual *activity of learning* with the support and enhancement of technology. The reasons for using the term TEL are similar to the reasons for using the term learning technology: it is an all-encompassing term that covers most aspects of learning with the support of technology, independent of the pedagogical approach. As with learning technology, the term TEL is used in the sense that it limits the use of technology to the use of ICT, excluding analog technology, such as radio, videocassette recorder (VCR), telephones, and so on. The term covers special cases where TEL can be used, such as for distance learning, flexible learning, blended learning, etc.

Examples of other terms that are sometimes used synonymously with TEL are *Computer-Based Training (CBT)*, *Computer Assisted Instruction (CAI)*, and *e-learning*. In this thesis, those terms are regarded as specialized directions that are all covered by the term TEL. The expression *ICT and learning* is commonly used, especially in the Nordic countries (see, e.g., Söderlund, 2000), when addressing TEL in schools and primary/secondary education. ICT and learning is about as broad as TEL and is here regarded as synonymous.

1.7.3 Object – as in learning object

The term ‘learning object’ is used when learning objects are addressed as a phenomenon and concept. The term is used in the same sense as a large part of the TEL-community uses the

term, that is, addressing the concept of platform and context independent, self-contained pieces of reusable digital learning content, which can be assembled to form a larger unit of digital learning.

The notion of an *object* is quite different when addressing learning objects compared to an object in *Object-Oriented programming* (OOP). An object in OOP is a clearly defined instance of a class with explicit properties, operations and attributes. Learning objects on the other hand are actually more similar to software components, but often without “*contractually specified interfaces and explicit context dependencies only*” (see the quotation of Szyperski (2002c), in chapter 1.2).

1.7.4 Digital learning resources

A *digital learning resource* is an arbitrary digital resource that is used for learning, or a digital resource intended to be used for learning. The reason for using the term digital learning resource is that the term learning object is often misused, as discussed in chapter 3, which easily leads to misunderstandings and confusion. The term learning object is still too vague and undefined, and is therefore merely used in contexts where learning objects are discussed as a phenomenon, while the term digital learning resources is used when such resources (including learning objects) are discussed in a general context.

1.7.5 Component

The term *software component* as defined by Szyperski (2002c) will be used when for addressing components from a software architecture and software development point of view. The term *component* will be used in a general sense to describe a constituent part (module) of a composite and modular architecture or infrastructure.

2 Projects and paper summaries

This chapter describes the projects in which the studies presented in this thesis were carried out. The purpose is to establish a better understanding of the context and background. This chapter also contains descriptions of my contributions to the appended papers in relation to my co-authors and paper summaries.

2.1 Projects

The studies and data collection were carried out within three different projects. These were three separate projects but related to each other in different ways, and their existence depended on each other. Altogether the outcomes of the projects make an important contribution to the research field and to this thesis by addressing research questions 1 and 2, presented in chapter 1.3, in suggesting a system-design approach that promotes modularity and unifies the predominating model for digital learning content, the learning object model, with a corresponding modular architecture-model for the virtual learning environment. Research question 3 was addressed through the suggestion and implementation of a prototype of a semantic web-based annotation model for representing pedagogical processes through dynamic linking of curricula and components in a modular learning architecture.

These projects and the studies presented in this thesis should be regarded in the light of my background of being a computer scientist on the one hand and being a teacher, working with teacher training, on the other hand.

2.1.1 ICT and schools – a complicated relationship

The *Virtual Workspace Environment (VWE)* project, described in chapter 2.1.2, has provided the empirical backbone of this thesis.

The VWE-project started in 1998, and was born out of a desire to change the way in which virtual learning environments were perceived. ICT seemed to be apprehended differently in many educational institutions (especially in schools) than in other sectors of society. This is, for example, indicated by Papert (1994), as well as by Pedersen (1998). This phenomenon revealed itself in the way that teachers often responded to the introduction of ICT as yet another problem, rather than the solution to pedagogical problems, or at least being a supporting tool for pedagogical practice and learning.

Figure 2-1 illustrates this situation. The figure was originally designed for a presentation (Paulsson, 1998) preparing for the initiation of the European Schoolnet (EUN) in 1998.

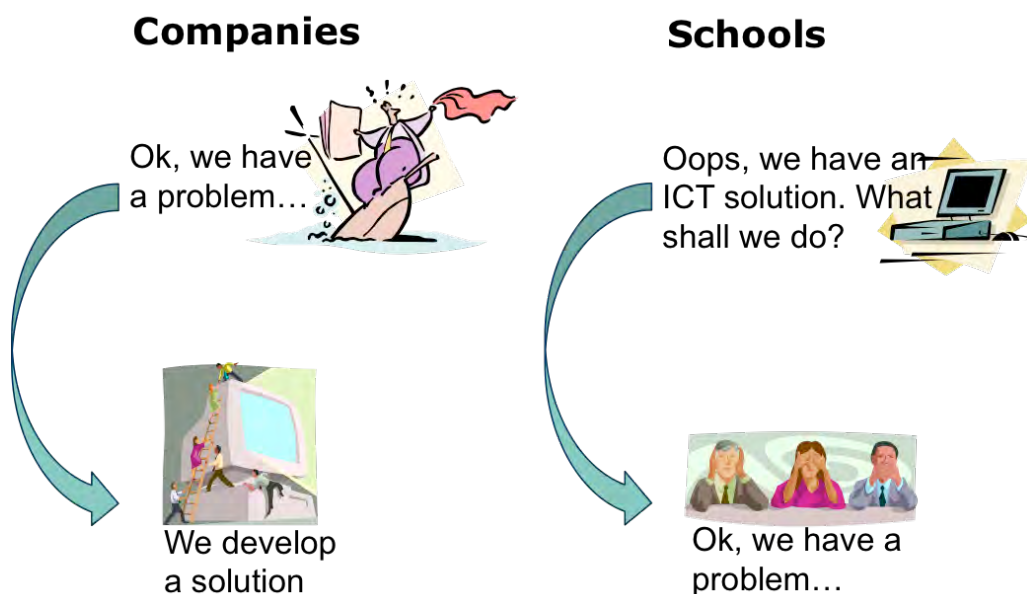


Figure 2-1. The view on ICT of many teachers, as understood in 1998. Unfortunately, this figure is still valid in many respects.

In Sweden, teachers also had (and still have) a rather complicated relationship to the use of ICT. This complicated relationship is essentially related to previous experiences from national ICT programmes and ICT projects. Söderlund (2000, p.9, pp. 48-50) examines the long-term relationship between schools and ICT in detail. Söderlund discusses how ICT has been introduced in Swedish schools between 1980 and 2000, and how teachers have understood ICT. Söderlund also points out that the adoption of technology is much slower in schools than in other parts of society. According to Söderlund, one reason for this may be that ICT has a strong symbolic value and the school system has been the subject of a number of politically initiated measures. The purpose and objective of the chosen political measures have not always been clear, except for the sense that there has been a belief that the use of ICT will lead to positive change. The result has often been that ICT has been introduced and implemented based on political conditions rather than on the conditions of teachers and students, and this is also likely to be one of the reasons for the attitude illustrated in Figure 2-1. Those circumstances are emphasized by Söderlund (2000, pp. 179-183), where he discusses the difference between the “political” approach described above, and the bottom-up approach taken in some cases at the local level, where ICT has been implemented based on the conditions and needs of pedagogues in combination with knowledge building for teachers.

Hence, this complicated relationship between pedagogues and ICT together with the top-down perspective seem to have been strong influential factors for the common belief that the design and implementation of ICT and ICT tools is something that cannot (and according to some - should not) be influenced by active involvement by teachers or students. There is an obvious risk that this may strengthen the attitude illustrated in Figure 2-1. The VWE project, described in the following chapter, was set out to involve pedagogues in the design and implementation of ICT.

2.1.2 The Virtual Workspace Environment³

The VWE project was aimed at changing the view on ICT, described above, by shifting some of the control to the users, that is teachers and students, and at the same time making it easier for teachers and students to participate in the actual design and implementation of their VLE and the ICT tools that they are supposed to use. This was motivated by an expectation that it would allow teachers and students to set the boundaries for the pedagogical activities to a higher extent, rather than allowing the technology to do that.

The basic idea of the VWE project was to develop a modular VLE that could be controlled and managed by teachers and students to a much higher extent than what was (and still is) usually the case.

The VWE philosophy was inspired by concepts and ideas from component technology in computer science and ideas similar to the ones leading up to learning objects, while maintaining a strong focus on modularity, adaptability and reuse. The rationale behind the VWE project was the assumption that modular content, that is learning objects, are not enough to obtain the level of adaptability and flexibility needed to support the variety of learning contexts, pedagogical approaches and the diversity of needs of teachers and students. Learning objects only contribute to the adaptability and flexibility of digital learning content, but the pedagogical freedom is still restricted by the VLE. This is a problem that relates to research questions 1 and 2 that were presented in chapter 1.3.

The idea constituting the basis for the VWE is that teachers and students should be able to *compose* and *aggregate* (as well as *disaggregate*) their personal and shared VLEs. The composition of a VLE is carried out using a “toolbox” containing components, in a similar way as content is supposed to be aggregated and disaggregated using learning objects.

Such modular concepts lead to a level of flexibility and adaptability that allows teachers and students to better adapt and form their (personal and shared) VLEs to their own needs, as well as to the needs of a group. As a consequence, this also affects the ability to adapt to the needs of a specific pedagogical approach, or a specific learning situation. An important condition for this kind of adaptability is the possibility of dynamically adding, as well as removing, functionality.

2.1.3 The Standardized Contextualized Access to Metadata (SCAM)⁴

The need for a general-purpose system for managing advanced metadata was born out of needs that arose within the VWE project, together with experiences and need from other projects at the *Royal Institute of Technology (KTH)*, and the *Knowledge Management Research group (KMR)*⁵.

The more modular the VLE and the digital learning content became, the greater was the need for sophisticated metadata management. The need for metadata that kept things together grew as the VLE transformed into a complex web of loosely coupled and (in terms of functionality) simple components, which interacted in a rather complex system of loose and dynamic relations. It was realized that *machine-readable* and *machine-processable* semantics would be needed in order to not put too much cognitive burden on the users, and to make it possible to locate components for different purposes and within different contexts, as well as for

³ The VWE project was funded by the European Commission, and by the Swedish National Agency for Education as a part of the European Schoolnet (EUN) project. The VWE project was lead and hosted by the department of Interactive Media and Learning (IML) at Umeå University.

⁴ The SCAM project is an ongoing project that is hosted by IML at Umeå University together with the Knowledge Management Research Group (KMR-group) at the Royal Institute of Technology (KTH).

⁵ <http://www.kmr.nada.kth.se/>

providing the VLEs with the potential to react, change and evolve based on feedback from components and actors within the system. This includes VLEs that are adaptive as well as adaptable. A VLE can be made adaptable by modularization, but it became apparent that machine-processable metadata were needed in order to also achieve *adaptivity*⁶.

The SCAM system, which is a result of the SCAM project, is an RDF repository that can be plugged in for use by (or as a part of) other software or services on the Semantic Web (Paulsson, 2003). SCAM was originally intended to be used as a development platform and code library for developers, but has over time evolved into two branches of stable, industry-strength applications as well. These are packaged as SCAM Nimble, for the development of *learning object repositories* (LOR) that conforms to standards such as Dublin Core (DCMI, 2004), IMS Metadata (IMS, 2001) and IMS Content Packaging (IMS, 2004), and the SCAM ePortfolio⁷. The SCAM ePortfolio is an ePortfolio for the Semantic Web, which can be integrated into VLEs as a service.

SCAM is currently distributed using a combined GPL/LGPL Open Source License (OSI, 2007).

Issues related to SCAM and the storage and management of RDF in relational databases are discussed in detail by Palmér, Naeve, and Paulsson (2004). However, these issues are outside the scope of this thesis.

2.1.4 The Annofolio project⁸

The Annofolio project can be regarded as a sub-project of the SCAM ePortfolio project. The objective of the Annofolio project was to explore the possibilities of creating dynamic relations, based on machine-processable semantics, between digital resources, such as information objects, curricula, learning object and other components in a modular learning architecture.

A prototype application, called *the Annofolio*, was produced by the project. The Annofolio is basically an ePortfolio with sophisticated annotation functionality, implemented as an add-on to the SCAM ePortfolio. The Annofolio was tested in a case where learning objectives and central concepts in the national steering documents were dynamically connected to learning objects in a LOR. The connections were made using semantic relations in the respective set of annotations that were added using the Annofolio tool (see paper 6 for detailed examples). This kind of semantically-based interlinking adds a certain amount of adaptivity, where digital learning resources can be found and explored based on the relation to corresponding learning objectives in the curriculum, expressed through semantic web annotations. Research question 3, presented in chapter 1.3, was addressed in the Annofolio project through the development of a prototype for connecting digital learning resources to the curriculum.

⁶ I make a distinction between *adaptable* and *adaptive*. Adaptable means that the VLE can be adapted through active actions by users, while adaptivity means that the VLE can automatically adapt and react to changes or inputs that occur in the VLE based on generated data, external input, learner preferences, attention metadata, and so on.

⁷ Also called Confolio.

⁸ The Centre for IT in Northern Sweden (CINS) hosted the Annofolio project, together with IML at Umeå University. The Annofolio project was funded by the Swedish Agency for Innovation Systems (VINNOVA) as a part of a project called Tools and methods for markup of educational content, lead by the research programme Digital Literature at Uppsala University.

2.2 Paper summaries

2.2.1 *My contributions to the papers*

I am the main author and have written all of the papers that are appended to this thesis. I have also been responsible for and carried out the major part of the scientific work presented in the papers.

The co-authors' roles have mainly been to assist with the implementation, design and coding of prototypes, which are indeed valuable contributions to the projects described above.

The work presented in the papers was carried out within the projects described in chapter 2.1. I have worked as the project leader in all three projects.

The nature of my contributions is presented below. I have, for the sake of clarity, chosen to explicitly state the contributions of each co-author as well. Observe that the co-authors have agreed to the descriptions below.

Paper 1. I wrote the paper and was responsible for carrying out the scientific work, including the presented studies that resulted in the suggestion of six areas for quality criteria for learning objects. I presented the paper at the eChallenges 2006 conference in Barcelona. Ambjörn Naeve assisted me with feedback and suggestions during the study and assisted in delimiting and formulating the six areas for quality criteria.

Paper 2. I wrote the paper and was responsible for the scientific work that resulted in the VWE Learning Object Taxonomy, which was also the main result presented in the paper. Ambjörn Naeve provided feedback and suggestions regarding the framing of the VWE learning object taxonomy. He also assisted in modeling and visualizing the VWE learning object taxonomy using the Conzilla concept browser⁹. The first VWE prototype and SOA framework were also briefly presented and discussed in paper 2. My contributions to that work are described below.

Paper 3 and paper 4. I wrote the papers and was responsible for carrying out the main part of the scientific work as well as for the development of the VWE concept. I was responsible for the design of the VWE SOA architecture framework as well as parts of the Java RMI-based SOA-architecture. I also contributed to the coding of the prototypes and development (and adaptation) of VWE Tools. Mikael Berglund contributed to the VWE project with substantial amounts of code and he had the main responsibility for the implementation design of the Web Services architecture.

Paper 5 and paper 6. I wrote the papers and was responsible for executing the main part of the scientific work. I was the originator of the Annofolio markup concept, which was first described in, (Paulsson, 2002). Jonas Engman, at CINS, made important contributions to the transformation of the Annofolio concept into a working method. He also contributed to the system design and the further development of the markup method, together with me. Engman also contributed with substantial amounts of code for the prototype. I contributed to the coding as well. My main focus was on the code related to XML and XSLT, and the integration with the SCAM ePortfolio. I carried out the study and the scientific work connected to the markup of the national steering documents, presented in paper 6.

⁹ <http://www.conzilla.org/>

2.2.2 Paper 1: Establishing technical quality criteria for Learning Objects

Published in the proceedings of the eChallenges 2006 conference (25-27 October 2006), in Barcelona, Spain.

Paper 1 presents a technical evaluation of learning objects from two Swedish learning object repositories. The paper also reviews some of the literature and research that addresses learning object models and definitions in order to propose a number of areas for working with technical quality criteria for learning objects.

The absence of common low-level definitions and abstract models for learning objects presents a threat to the interoperability and technical quality of learning objects - as well as to learning objects as a concept. Most of the vision associated with learning objects has yet to be fulfilled, because most current implementations of learning objects do not fulfill this vision. It is therefore essential to establish criteria that secure the technical quality of learning objects as one step in addressing those issues. As a result, six areas for technical quality criteria were suggested.

The quality criteria were tested as a basis for the VWE learning object taxonomy, and by implementation in the Virtual Workspace Environment (VWE), described in paper 2, paper 3 and paper 4.

2.2.3 Paper 2: Virtual Workspace Environment (VWE): A Taxonomy and Service Oriented Architecture Framework for Modularized Virtual Learning Environments - Applying the Learning Object Concept to the VLE

Published in the International Journal on E-Learning, 5(1), 45-57, (2006).

The term *learning object* is used for referring to digital learning resources that are supposed to be modular, fairly context independent and have a high level of reusability. The basic idea is that several learning objects can be assembled to form a larger pedagogical unit (a learning module) that can be used for a specific learning context. Even though learning objects have been around for more than a decade, and the concept is widely used and implemented, there is no common or broadly accepted definition.

The atomic learning object taxonomy, suggested by Wiley (2002a), works fairly well for defining the characteristics of learning objects at some levels; especially on the functional and usage levels. Wiley's taxonomy is based on the level of complexity and interactivity of learning objects, but it fails to define learning objects at the architectural and interface levels, i.e., how to actually accomplish the functional characteristics that are expressed in the taxonomy.

An alternative abstract model and learning object taxonomy (*the VWE learning object taxonomy*) was presented in this paper. The VWE learning object taxonomy is compatible with Wiley's atomic learning object taxonomy, which means that it can essentially be regarded as a complementing taxonomy rather than a competing taxonomy. All the taxonomy levels of Wiley's taxonomy can be implemented using the VWE learning object taxonomy, and the objective of the VWE learning object taxonomy is to add an abstract model for learning objects where data, application logics, and presentation are separated into different layers. The VWE learning object taxonomy does not only address modular content, it also addresses functionality components in VLEs using the same modular concept as learning objects normally represent for content. This becomes possible when application logics, data, and presentation are separated so that a learning object can represent application functionality as well as digital content independently of each other.

Even though this paper briefly describes the VWE SOA architecture as well, the major contribution of this paper is the VWE learning object taxonomy.

2.2.4 Paper 3: Implementing a Service Oriented Architecture for modularized Virtual learning Environments: a comparison of two approaches

Published in the International Journal of Web-Based Learning and Teaching Technologies, 3(1), (2008).

This paper compares the implementations of two different Service Oriented Architecture approaches for modular virtual learning environments. Both prototypes implement a similar system design, using two different approaches: the first prototype is implemented using Java and *Remote Method Invocation (RMI)*, while the second prototype is implemented using Java and web service technology. The objective of the system design was to implement an architecture that corresponded to the VWE learning object taxonomy in order to provide an architecture framework that facilitates the implementation of modular virtual learning environments.

Some design issues are discussed in the paper together with a comparison of the implications of using Java RMI versus web service technology. The comparison is made from the point of view of modular VLEs, and the main purpose is not to compare general technical qualities, such as differences in performance, but to demonstrate that the architecture framework can be implemented using different approaches and technology platforms, and to point out similarities and differences, as well as advantages and disadvantages.

The comparison showed that some minor changes had to be made in the system design when moving from Java RMI to web services. The results were in favor of the web service approach, although it is technically inferior in some ways and trade-offs were made in terms of performance and efficiency. The advantages of web services are especially related to the use of web service standards and the use of the http-protocol together with the use of an XML-based protocol, i.e., SOAP. However, some of the biggest disadvantages are also related to the use of the http-protocol and to the use of an XML-based protocol, as the combination of http and XML create overhead that influence several layers in the architecture.

The most important results presented in paper 3 are that the prototype implementations of the VWE learning object taxonomy proves that an approach where digital learning content and the VLE are incorporated into the same abstract model works, and that it can be implemented using different technology frameworks.

2.2.5 Paper 4: A Service Oriented Architecture-framework for modularized Virtual Learning Environments

Published in the proceedings of the m-ICTE2006, (22-25 November), in Seville, Spain

Paper 4 is a conference paper that describes the final VWE web service-based architecture in detail. The web service architecture, which was compared to the Java RMI approach in paper 3, was the final outcome of the VWE-project.

Even though papers 3 and 4 have some overlap, they each have a different focus. Paper 3 compares two approaches focusing on the comparison and on showing that the VWE learning object taxonomy can be implemented using several approaches. Paper 4 focuses on the resulting web service architecture, which was also, considering the current state of SOA, suggested as the most reasonable approach to implement modular VLEs in accordance to the VWE learning object taxonomy.

2.2.6 Paper 5: Treating metadata as annotations: separating the content mark-up from the content

Published in the International Journal of Emerging Technologies in Learning (iJET), 1(3), 2007.

Structural markup and metadata are becoming increasingly important as the use of digital learning content increases. Metadata are needed for discovery of digital learning resources as well as for describing different aspects of digital learning resources, such as intended use, technical constraints, relations to other resources, context of use, and so on.

Structural markup, commonly XML-syntax, is needed to describe the *structure* of the digital learning content. Structural markup can be used for presentation, transformation, and for processing of digital learning content in different ways. There are, however, increasing needs to add metadata and annotations with machine-processable semantics that describe the internal structure of a resource. This way, parts of a digital resource can be described in the same way as metadata are used to describe the entire resource in a repository. In many cases, only a small part of a resource is of interest, such as an excerpt of a text or a clip from a movie. It is sometimes desirable to add annotations that emphasize a certain perspective, a certain usage, or in other ways comments on a part of a resource. This can, for example, be used to describe relations to other resources or to other sources of information, as is discussed in paper 6.

Traditionally, markup languages such as *Standard Generalized Markup Language (SGML)* and XML have been used for describing both structure and semantics, often combined in the same schema, such as in the *Text Encoding Initiative (TEI)* ¹⁰, which is one of the most used schemas for annotation of literary and linguistic texts. This mix of syntax and semantics is problematic for several reasons: firstly, XML was developed for expressing *syntax and structure not semantics*, secondly it is not possible to build a *metadata ecology* (M Nilsson, Palmér, & Naeve, 2002) on a principle where metadata have to be added inside a resource internals, that is into the XML-file in the case of TEI. Besides causing problems concerning access and rights, the XML-file would simply become too messy and grow uncontrollably as new metadata are added. There is no discernible realistic way of controlling the integrity of the added markup, as anyone would be able to edit the XML-file. In addition, adding descriptive semantic data into XML-files creates an unwanted processing overhead.

The problems described above were addressed by developing a method and a set of corresponding tool-prototypes for “pointing-in” annotations into the right section of an XML-file. Annotations are represented using RDF, which (in theory) means that any metadata-model or *application profile*¹¹ can be represented and used for markup. Metadata are stored externally, separated from the resource, in a SCAM-repository. Anyone can add metadata and in the case of the Annofolio, metadata are stored in user portfolios in the SCAM ePortfolio-system. The method creates a layered and ecology-like metadata system, where different layers of metadata are separated, stored and managed at an individual level by the provider.

The purpose of the Annofolio method is to use the structural markup of XML-resources to “point-in” descriptive annotations that are stored in the SCAM-based Annofolio ePortfolio.

The main findings presented in paper 5 are a general-purpose method and architecture for adding annotations to existing XML resources. The method relies on existing technologies, such as XML for structure and syntax, *XPointers* for identifiers, the *eXtensible Stylesheet*

¹⁰ <http://www.tei-c.org/>

¹¹ In a metadata context, an application profile is a defined set of elements and guidelines for metadata for a specific application that can be put together by combining one or more metadata schema.

Language (XSL) for XML transformation, the *Document Object Model* (DOM) for managing the hierarchy of XML-documents, JavaScript for manipulation, and semantic web technologies (RDF and Annotea), for expressing machine-processable semantics.

2.2.7 Paper 6: Marking the National Curriculum - A New Model for Semantic Mark-up

Published in the proceedings of the eChallenges 2005, (2005, 19-21 October), in Ljubljana, Slovenia.

Paper 6 describes a case where the Annofolio was used to add annotations to the Swedish national steering documents¹². The method, which was described in detail in paper 5, was developed as an alternative to XML-based markup methods, such as the *Text Encoding Initiative* (TEI), where metadata semantics are expressed within the XML-document

Parts of the Swedish national steering documents were annotated using the Annofolio-prototype. Central concepts and learning objectives were furnished with machine-processable metadata in a selected number of course plans. A basic layer of curriculum metadata was provided by indexers at the Swedish National Agency for Education in order to form the basis for a *curriculum service* that was built based on the course plans in the national steering documents.

Two demonstrators were developed to illustrate how the machine-processable metadata and the curriculum service can be used. The first demonstrator used the markup tool for compiling *digital compendiums*, consisting of excerpts from course plans that were chosen by the user by using annotations to mark the starting points and end points of the sections of interest. The markup was used as the basis for the generation of a digital compendium using the *eXtensible Stylesheet Language: Transformation* (XSLT)¹³. The second demonstrator illustrated how the annotated steering documents could be dynamically linked to learning object metadata. The later presumes that learning object metadata are expressed using RDF (cf. Nilsson, Palmér, & Brase 2003). Suitable learning objects were located based on the semantics expressed by the annotations describing, for example, a specific learning objective within a course plan. In addition, the connection could also be made in the opposite direction; that is, metadata associated with a learning object could be used to locate relevant sections in the steering documents.

The architecture of the Annofolio-system is briefly described in this paper. However, the main relevance, in the context of this thesis, is to illustrate how the Annofolio method can be used to generate semantic relations between curricula and digital learning resources; that is between pedagogical instructions and other components in the learning environment.

In the following chapter I will present my theoretical framework. The theoretical framework will serve as the point of departure where my research results and my contributions to the research field are discussed in chapter 4 and 5.

¹² The Swedish national steering documents (<http://www3.skolverket.se/ki03/front.aspx?sprak=EN>) are the Swedish national curriculum, regulations, laws relating to public education, and course plans.

¹³ XSLT is a stylesheet language designed for transforming the structure of XML documents. XSLT is also used to transform XML documents into other file formats, such as html.

3 Research field and theoretical frame of reference

This chapter describes the research field and the theoretical framework in which the research presented in this thesis was carried out. Because the focus of this thesis is on modularity and the development of virtual learning environment, this chapter will span both pedagogical theory and theory from computer science. This width, involving several research fields, is also the reason why this chapter may appear as unusually extensive. The purpose is to illuminate the interrelations between technology and pedagogy.

3.1 Theoretical perspectives

The research presented in this thesis concerns the interrelations between learning theories, instructional design¹⁴, and the role of modularity in the design and development of learning technology. The understanding of these relations is important for the understanding of TEL as well as for the understanding of many of the issues involved in the design and development of VLEs. The use of ICT is, in general, becoming more and more applied and process focused, and by that also more interdisciplinary. TEL is a typical application field of ICT and it is important to regard TEL and learning technology from a holistic point of view. Dodig-Crnkovic, and Crnkovic (2003) express this point of view from a general perspective:

“The development of human thought parallel to the development of human society has led to an emergence of Sciences that do not belong to any of the classic types we have described earlier, but rather share common parts with several of these. Many of the modern Sciences are of interdisciplinary, eclectic type. It is a trend for new Sciences to search their methods and even questions in very broad areas.” (p. 1)

Even though this thesis focuses on the technological (computer science) aspects of TEL and the design and development of VLEs, the reasoning above implies that a synthetic worldview is needed in order to understand the issues raised in chapter 1, and in order to address the research questions formulated in chapter 1.3.

My theoretical framework needs to be rooted in several research fields, spanning from psychology and pedagogy to computer science. I have adopted an eclectic approach to my research in order to cope with this.

The objective of this chapter is not to explain different theoretical standpoints in detail, but to increase the understanding of, and to some extent establish the relations (the main threads and connecting thoughts) that more or less explicitly exist between pedagogy and technology. These are relations that are sometimes overlooked or not understood. The focus is on the mainstream theories, and I have taken the freedom to focus on the aspects that are of special relevance for this thesis, well aware that many aspects of the theories are excluded. The reason for this is that all of the theoretical fields are extremely extensive and delimitations need to be done. In spite of this it is my intention to illuminate a couple of different viewpoints within the different fields. It should also be noted that some of the phenomena described in this chapter are not scientific theories in a literal sense, but are nevertheless relevant for this thesis and for establishing my theoretical framework.

I have structured this chapter into three parts: *pedagogical theory*, *TEL theory*, and *computer science theory*. This structure could be questioned for several reasons, some of the

¹⁴ I use the term instructional design in the first section (pedagogical theory) as it refers to an established research field. The term *learning design* will be used in relation to Educational Modelling Languages (EML) (see chapter 3.10), where it is appropriate. The term *pedagogical design* will be used as a general, collective term to address the activity of planning and designing pedagogical activities - including instructional design and learning design.

pedagogical theories, such as that learning theory could be argued to be a part of psychology. There are, of course, no clear boundaries between the three fields and there are overlaps and grey zones, especially in the TEL section, which is situated somewhere in-between technology and pedagogy, but this disposition suits the purpose of this thesis in terms of clarity and structure.

Pedagogical theory deals with theory that is of general relevance for learning and teaching, TEL theory is “in-between theory” that is not solely pedagogy, and not solely computer science or technology, but that has a clear TEL focus. Computer science theory deals with theories that are of relevance for the design and implementation of learning technology, in terms of software, software architecture and infrastructure - with a specific focus on modularity. In some cases I have chosen to emphasize the historical perspective. The reason for this is that I find the historical viewpoint relevant for determining and understanding the current position and state of TEL, as well as for understanding the origins of my research questions (chapter 1.3) and the emerging problems in the research field (chapter 1.2). This is also the reason why the ordering is chronological.

3.2 Learning theory

This chapter gives an overview of the learning theories that are indicated by the literature to have been the most influential during the 20th century. The purpose is to establish a better understanding of the relation between learning theories, instructional design, and the design of learning technology, together with chapter 3.3 and 3.4. Learning theory is also used as a tool to encircle some of the emerging problems described in chapter 1.2, as well as for the derivation of the origins of my research questions, in particular research question three.

The literature indicates, consistently, that the three dominating directions of learning theory during the 20th century were *behaviorism*, *cognitivism* and *constructivism* (see, e.g., Schunk, 2003; Ertmer & Newby, 1993; Mylov, 2000). Each learning theory has different implications for the implementation of pedagogical instruction (the result of instructional design) and by that also for the implementation of learning technology (see, chapter 3.4.1, chapter 4, and chapter 5).

The purpose of chapter 3.2.1 to 3.2.4 is to briefly discuss the origins and the main features of the different learning theories in order to establish a basic understanding of the similarities and differences of behaviorism, cognitivism, and constructivism.

3.2.1 Behaviorism

Learning theories based on behaviorism were dominating during the first 60-70 years of the twentieth century (see, e.g., Schunk, 2003; Saettler, 2004). The basis for behavioristic learning theories is the principle of stimulus-response. According to behavioristic learning theories, behavior is triggered by external and environmental stimuli. Skinner (1976) refers to this mechanism as operant conditioning. The difference between classical conditioning (Pavlov, 1927), and operant conditioning is mainly that operant conditioning addresses voluntary behavior and is thereby better applicable to human learning. Skinner and Pavlov are likely to be two of the most well-known behaviorists and their respective work has been influential in the field of behavioristic learning theory. Many commonalities can be found in the work of Skinner and the work of Watson, such as their view of conditioning (see, e.g., Walker, 1990). Watson is, together with Thorndike, often regarded as two early and influential behaviorists.

Thorndike established a behavioristic branch called Connectionism (e.g., Walker, 1990). He regarded the mind as an interconnected network of small and simple units that are associated

to each other. Much of his work was carried out in relation to testing the intelligence and learning ability of children. Thorndike formulated three basic laws that established the relations between stimulus and response (e.g., Walker, 1990): 1) *the law of effect*, which states that positive response will enhance behavior and negative response will usually weaken behavior; 2) *the law of readiness*, which states that a series of responses can be chained together in order to achieve goals; and 3) *the law of exercise*, (which was later withdrawn) which claims that the connection between stimulus and response is that behavior is established by repeated connections of stimulus and response. Like Pavlov (1927), Thorndike mainly based his research on animal behavior.

Watson transferred his research based on animal experiments to human learning and behavior as well. He argued that psychology should only pay attention to observable behavior and not worry about mental processes (Andersson, 2000, pp. 63-75). Watson (1920) even carried out an experiment involving a child (Albert) with the purpose of proving that classical conditioning (conditioned fear in Albert's case) was transferable to humans. Watson (1920) meant that all humans are born with a couple of basic reflexes, and that the rest of our knowledge and behavior is the result of stimulus-response. Watson is sometimes referred to as the father of behaviorism, and is maybe most famous for his idea of being able to shape children to whatever he wanted.

"Give me a dozen healthy infants, well-formed, and my own specified world to bring them up in and I'll guarantee to take any one at random and train him to become any type of specialist I might select—doctor, lawyer, artist, merchant-chief and, yes, even beggar-man and thief, regardless of his talents, penchants, tendencies, abilities, vocations, and race of his ancestors." (Watson, 1925)

According to pedagogical literature, Skinner is the researcher who is most commonly associated with behaviorism from a learning theory point of view. Even though Skinner seems to have based much of his research on the work by Watson, Skinner studied the stimulus-response pattern from a human learning perspective, establishing the principle of operant conditioning (see, e.g., Andersson, 2000, pp. 63-75). The basic mechanisms of operant conditioning differ from the classical conditioning in that learners operate on the environment and are rewarded for the right behavior with some form of feedback (*reinforcement*) that can be either positive or negative. Skinner (1976) believed that no reinforcement leads to that certain behavior is not repeated and that punishment can repress behavior. Skinner used the concept of chaining to explain how small learning steps could be chained together as sequences in order to learn sequences of behavior. Chaining later became one of the corner stones of instructional design (see chapter 3.4), and is sometimes referred to as *chunking and sequencing*.

A distinguishing characteristic of behaviorism is that the learner is given a rather passive role in the learning process: the learner is an object for teaching (see, e.g., Andersson, 2000). The learner is viewed as *tabula rasa* (i.e., blank state), an expression coined by the 18th century philosopher Locke (see, e.g., Nilsson, 2001), which implies that humans start out with no mental content. Knowledge is gained by modifying behavior by stimulating change of behavior, using positive and negative reinforcements. The mind is a black box and cognitive processing is generally not considered in the learning process.

Behavioristic learning theories provided a distinct and clear basis for the early attempts of using instructional design with technology in that it was – and still is – quite straightforward to represent using technology. According to behaviorists, every step in the learning process can be predicted and by that described and programmed beforehand. There are no individual differences or unexpected turns that need to be considered. Behaviorism was the dominating

learning theory during the main part of the 20th century. According to Mylov (Mylov, 2000), behavioristic learning theories started to be challenged first in the sixties, even though alternatives existed long before that. The first direction to challenge behaviorists' learning theories was cognitivism (see, e.g., Schunk, 2003, pp. 83-135; Saettler, 2004; Marton & Booth, 2000, pp. 25-29; Bjerg, 2000).

3.2.2 Cognitivism

A shift in favor of learning theories based on theories and models from cognitive science seemed to have occurred between the mid-fifties and sixties, (see, e.g., Schunk, 2003, pp. 83-135; Saettler, 2004; Marton & Booth, 2000, pp. 25-29; Bjerg, 2000). Partly, this shift seems to have been an evolvement of behaviorism that led to theories such as the *social learning theory*, described by Bandura and Walters (1963) and (cf. Bandura, 1977), and some regard social learning as the bridge between behaviorism and cognitivism (Schunk, 2003, pp. 83-135). Bandura and Walters (1963) argued that learning could be accomplished through modeling, and by observing and imitating other people, as well as through the traditional behavioral mechanisms. Bandura (1977) found the idea that the environment is the cause of all behavior to be too simplistic to explain the complexity of human beings. He argued that the environment and individual's behavior cause each other. Bandura claimed that an individual's personality is created out of interplay between the environment, behavior, and internal psychological processes. Bandura also found that learning could be achieved merely by watching others, emphasizing social aspects in the cognitive process (Bandura, 1986), also referred to as *Social Cognitive Theory* (see, e.g., Schunk, 2003, p. 83). This represents a view of learning that stands out as being quite different, and to some extent the opposite of the ideas represented by behaviorists, like Watson, Skinner, and others.

From the above point of view, cognitivism can be seen as a reaction to behaviorism – a reaction that moved the focus from observable behavior to more complex cognitive processes such as thinking, memory, and problem solving, by shifting the focus to internal cognitive processes (see, e.g., Snelbecker, 1974). The increased focus on cognitive processes was a result of the progresses within cognitive psychology¹⁵ that were made during the fifties, referred to as the *cognitive revolution* (see, e.g., Saettler, 2004, p. 318; Marton & Booth, 2000, pp. 25-29).

Within cognitivism, knowledge is regarded as a schema that is considered to be an internal structure of mental constructions. The schema changes when learning occurs. New information is compared to the learner's existing schema, which may be altered or extended when new knowledge is gained. Typical for cognitive learning theories is that the learning process is regarded as a three-stage information-processing model, where input is first registered by the learner's senses, then entered into the short-term memory, and finally transferred to the long-term memory for storage and retrieval, referred to as *information processing*, described by Schunk (2003, pp. 136-189). Hence, cognitive processing and previous knowledge are given a lot of attention in cognitive learning theories. Learning is regarded as being a mental activity that concerns the receipt, organization and storage of information (Jonassen, 1991c), also taking higher mental processes into account. Cognitivism changed the view of the learner from being passive to being active, even though the role of the environmental conditions was still emphasized. Clark (2006), summarizes cognitivism as “...learning is about active construction of new knowledge by interacting with new information, and instruction is about promoting the psychological processes that mediate that construction”. (p. 5)

¹⁵ Nowadays cognitive psychology is commonly referred to as being a part of cognitive science.

Examples of other influential contributors to cognitive learning theory are Piaget, introducing the *Stage Theory of Cognitive Development* (Piaget, 2006; Piaget & Inhelder, 1997) (see also chapter 3.2.2), Bandura, (see above), Merrill (1983), with the *Component Display Theory* (CDT) Reigeluth (1999b), who introduced *Elaboration Theory* and Bruner (1961), whose theory of *Discovery Learning*, an inquiry-based instructional method that emphasizes personal discovery of knowledge, which are sometimes regarded as a bridge to constructivism (see, e.g., Schunk, 2003, pp. 241-284).

The pedagogical change of focus, from behaviorism to cognitivism, influenced the practice of instructional design (Ertmer & Newby, 1993), which in turn had implications for the representation of instructional designs and pedagogical processes. For those reasons, the work of Merrill and Reigeluth is further discussed in chapter 3.4. This also led to new instructional models that were set out to adapt pedagogical instructions to cognitive learning theories for use with TEL. One such example is the *Inquiry Teaching model* by Collins (1977), which was a question-based model, used within *Computer Assisted Instruction* (CAI) (see also chapter 3.5).

For the understanding of learning theory, I find it important to emphasize that there is a difference between cognitive science and cognitive learning theories. While cognitive science is the science practiced within psychology, cognitive learning theory (cognitivism) is the application of theories from cognitive science, addressing a specific branch of epistemology. The reason that this distinction needs to be made is that cognitive science is important for other learning theories as well, such as for constructivistic learning theories that eventually challenged cognitivism.

3.2.3 Constructivism

Simpson (2002) argues that constructivism is a psychological and philosophical perspective, and an epistemology rather than a learning theory. However, constructivism is often referred to as a learning theory in line with behaviorism and cognitivism (see, e.g., Schunk, 2003, pp. 285-328).

Compared to cognitivism, constructivism emphasizes personal experience, and learning is best accomplished by doing (see, e.g., Dewey, 2004). The role of the teacher tends to transform into a guiding and mentoring role, as knowledge transfer from teachers to students is not considered to occur. As with other learning theories, the roots of constructivism can be traced back in history. Rousseau, for example, criticized that children were taught *about* things, rather than *experiencing* the things themselves. Rousseau said that: “*You think you are teaching what the world is like; he is only learning the map*”, cited by (Subramaniam, 2002). Rousseau argued, like modern constructivists, that the emphasis should be on learners’ *construction* of their own mental models, rather than on teachers conveying theirs. Like Piaget, Rousseau devoted his attention to the developmental stages of children. He introduced a theory that explained the development of a child in four stages (see, e.g., Egidijs, 2003, pp. 17-24). Rousseau’s *stage theory* was later revised by Piaget (2006) and presented as the *Stage Theory of Cognitive Development*.

Schunks overview (2003, pp. 285-328) shows that the individual construction of knowledge is one of the core values of constructivism, and knowledge is regarded as something that is individual in the sense that everyone constructs his or her own representation of the world and there is no objective truth. Learning occurs in a context and the more authentic the context is the more efficient learning becomes, and in social constructivism the social context is especially emphasized, “*Constructivism highlights the interaction of persons and situations in the acquisition of skills and knowledge*” (Schunk, 2003, p. 287). The active nature of learners

is strongly emphasized by constructivists and by necessity these are assumptions that have a strong influence on didactics, instructional design, and pedagogical methods. They also affect the teachers' role in the way that they put the focus on creating and structuring situations that stimulate learners to actively engage with content and act in a social context, rather than traditional teaching. The teacher becomes a mentor and is expected to provide guidance rather than on acting in the traditional role of the lecturing teacher. According to Schunk (Schunk, 2003, pp. 285-328) this has also influenced curriculums toward synthetic and integrated approaches.

Constructivistic learning theories challenge the traditional, lecture and teacher-centered teaching approaches and put the learner and the construction of knowledge and personal mental models in the centre. Learning becomes a personal as well as a social process of assimilation and accommodation when the learners assimilate new knowledge into previous experience, changing mental models, testing them against social interactions, assimilating again and so on (Piaget, 2002).

Constructivism is sometimes divided into three main directions: *exogenous constructivism*, *endogenous constructivism*, and *dialectical constructivism* (see, e.g., Schunk, 2003, pp. 285-328). Exogenous constructivism refers to knowledge as a construction of the external world, endogenous constructivism refers to knowledge as being derived from previous knowledge, and is a cognitive abstraction of the external world, and dialectical constructivism refers to knowledge as the product of the interaction between the learner and the environment. Knowledge is influenced by the external world as well as by the mind (see, e.g., Schunk, 2003, pp. 285-328; Piaget, 2002).

Two of the major contributors to constructivism, as it is perceived today, are Piaget and Vygotsky (see, e.g., Schunk, 2003, pp. 285-328; Marton & Booth, 2000). Piaget's research showed that children learn from doing, not from being fed with information. He argued that children are testing their mental models against new experiences, and new knowledge occurs through assimilation and accommodation, in which new experience is assimilated into old mental models (Piaget & Inhelder, 1997). This triggers an accommodation process that changes mental structures to accommodate new experiences. Piaget meant that we constantly refine (construct) our mental model of the world, adapting it to new knowledge and experience. Piaget's work focused on children, and children's cognitive development. Even though Piaget is often considered to be a cognitivist (see, e.g., Egidius, 2003), his contributions to constructivism are important and I have, like Schunk (2003, pp. 285-328) and (Marton & Booth, 2000, pp. 21-29), chosen to discuss Piaget in the context of constructivism as well.

Vygotsky (1978) developed a *sociocultural constructivist theory* that emphasizes social and cultural factors in cognitive development. Vygotsky argued that social interaction is essential for the construction of knowledge, and that language is our most important instrument for this. Like Piaget, Vygotsky's research was mainly focused on the cognitive development of children. The fundamentals of Vygotsky's theory state that social interaction and its associated tools, are essential to learning, and that self-regulation is developed through internalization (Vygotsky, 1978), see also Meece (2002, pp. 169-170). Human development is a result of cultural transmission of tools (language and symbols) where language is the most critical tool related to the *zone of proximal development (ZPD)*, which is the difference between what can be done by the individual, and what can be done by help of others (Vygotsky, 1978).

Another central researcher within the field of constructivistic learning theory was Dewey, who is most famous for his contributions to a branch of constructivism called *pragmatism*.

Dewey coined the phrase “*learning by doing*”, a phrase that summarizes the core of pragmatism as a learning theory.

3.2.4 Comparing learning theories

A framework for comparing learning theories is needed in order to understand the differences and similarities of the learning theories that have dominated the twentieth century. Schunk, (1991, pp. 17-21), Schunk suggests five questions¹⁶ that can be answered based on different learning theories, and that help in illustrating and understanding the differences and similarities. Table 3-1 shows a simplified comparison, based on the work of Schunk (1991, pp. 17-21; Schunk, 2003, pp. 17-21), and the work of Ertmer and Newby (1993). Table 3-1 illustrates the key differences of each learning theory.

	Behaviorism	Cognitivism	Constructivism
<i>How does learning occur?</i>	- Learning is a change in observable behavior. - When a proper response to an environmental stimulus is given, learning occurs. - Reinforcement of responses increases the likeliness for a behavior to be repeated.	- Learning is discrete changes between states of knowledge. - The acquisition of knowledge is a mental activity that entails internal coding and structuring by the learner. - The learner is perceived as active in the learning process.	- By constructing meaning from experience. - The learner constructs his/her own reality interplaying with the surrounding world, based on experience and interaction. - Knowledge is not mind-independent and cannot be transferred to learners. - The internal representation of knowledge is under constant change and there is no objective reality.
<i>Which factors influence learning?</i>	- Environmental factor are emphasized. - The arrangements of stimuli and consequences is critical	- Environmental factors are important. - Mental activities that lead to a response are emphasized, as well as mental planning, goal-settings and organizational strategies. - Learners' thoughts, attitude and values are considered as important. - Encouraging learners to use appropriate learning strategies.	- Both learner and environmental factors are critical, with a focus on the interaction between those. - Learning contextual and content knowledge should be embedded in the situation where it is used. - Learning must occur in realistic settings using tasks that are relevant to the experience of the individual. - Learning is depending on activity, concept and culture.
<i>What is the role of memory?</i>	- Is not considered. - Forgetting is caused by non-use of a response over time.	- Learning occurs when information is stored in the memory and organized it in a meaningful way. - Forgetting is the inability to retrieve information from the memory. - Both the knowledge and the use of knowledge is stored in memory	- The memory is always evolving and changing. - Representations of experiences are not stored as formalized declarative knowledge. - Focus is on flexible use of preexisting knowledge rather than recall of pre-packaged schemas. - The focus is on the creation of cognitive tools in the sociocultural context of the individual.
<i>How does transfer occur?</i>	- Generalization causes transfer - Transfer occurs in identical or similar situations, across common elements.	- Transfer depends on how information is stored in memory. When a learner understands how to apply knowledge in different contexts, transfer has occurred. - Prior knowledge is used to establish boundary constraints for identifying similarities and differences.	- Learners create situation-specific understanding by assembling prior knowledge from multiple sources, supported by experience. - By involving authentic tasks in relevant contexts. - Transfer is facilitated by contextualization and experience. - By emphasizing the portraying of tasks rather than defining the structure of learning required.

¹⁶ I have used the 1st edition of *Learning Theories: an Educational perspective* (Schunk, 1991), as this is the edition that is used by Ertmer and Newby (1993). The reason is that the list needs to be consistent with the work of Ertmer and Newby that I use in chapter 3.4.1. In the latest edition (4th) (Schunk, 2003) the list has been changed to: *How does learning occur, which factors influence learning, what is the role of memory, what is the role of motivation, how does transfer occur, which processes are involved in self-regulation, and what are the implications for instruction?*

<i>What types of learning are best explained by the theory?</i>	- Simple learning involving discriminations, generalizations, association and chaining. - Basic skills training.	- Complex learning involving reasoning, problem solving and information processing. - Knowledge that can be sized and chunked so that it can be easily assimilated and/or accommodated.	- In constructivism, it is impossible to identify types of learning independent of content and learning context. - It is not possible to isolate units of information or knowledge domains according to hierarchical analysis of relationships. - Complex knowledge in ill-structured domains. - Knowledge at the advanced and expert level (Jonassen, 1991a)
---	---	--	--

Table 3-1. A comparison illustrating the key differences between learning theories. Based on the research by Ertmer & Newby (1993).

Ertmer and Newby (1993) point out that there is no single “best” learning theory. Different learning theories support different types of learning and learning objectives, and it depends on the situation, on the learners, and on the learning objectives which theory is the most suitable. In most cases the learning process is constantly changing, and within one learning situation more than one learning theory may be involved. Learning strategies may be subject to change as the learner progresses and new levels of knowledge are reached and the complexity increases (Shuell, 1990). Hence, different types of knowledge are best taught and learned in different ways. For example, facts and procedural knowledge are not thought of in the same way as knowledge and concepts that depend on problem solving, complex mental processes, and social interaction (see, e.g., Ertmer & Newby, 1993; Shuell, 1990; Schunk, 2003).

Each learning theory has its implications and limitations regarding what type of knowledge that can be targeted and how learning is accomplished. In the context of this thesis it is mainly the implications for instructional design and the representation of instructional design in technology systems that are of interest in order to address my research questions. Instructional design and the connection between instructional design and learning theories are discussed in chapter 3.4.

As discussed above, different learning theories support and explain different types of learning and learning objectives, which imply that there may be a need to classify knowledge and learning objectives. This is discussed in the following chapters.

3.3 Classifying knowledge and learning objectives

There are several different ways of describing and classifying knowledge and learning objectives. Such classifications can be used as a tool to analyze the implications of learning objectives for the use of learning theories, instructional design strategies, and the design and development of learning technology. Jonassen (1991a) made such a classification where he describes three categories (levels) of knowledge: *introductory*, *advanced* and *expert levels*. Jonassen’s categorization may seem a bit rough, but can be useful to describe the connection between learning theories and their pedagogical implications at a general level, such as in Table 3-1. However, for more in-depth analysis there is a need for sharper tools, and one such tool is *Bloom’s taxonomy* (Bloom & Krathwohl, 1956), or the revised version of Bloom’s taxonomy that is more focused on teaching and learning practice (Anderson & Krathwohl, 2001). In this thesis, Bloom’s revised taxonomy is used for illustrating the connections between learning theories, different types of knowledge and learning objectives, learning design, modularity, and the system representation of instructional design in learning technology.

3.3.1 Bloom's Taxonomy of Educational Objectives

In 1956, Bloom and Krathwohl (1956) published their *Taxonomy of Educational Objectives for the Cognitive Domain*, commonly referred to just as *Bloom's taxonomy*. Bloom's taxonomy has been (and still is) influential as a classification schema for educational objectives. It is also used as a reference and tool by a wide variety of pedagogical practitioners (Anderson & Sosniak, 1994). Bloom's taxonomy has made it possible for instructional designers to better analyze and specify instructional outcomes and objectives.

The basic idea of Bloom's taxonomy (Bloom & Krathwohl, 1956) is to arrange educational objectives in a hierarchal structure. The taxonomy is divided into three domains: *Affective* (attitude), *Cognitive* (knowledge), and *Psychomotor* (skills). It is mainly the cognitive domain that is of relevance for this thesis. According to Andersson and Sosniak (1994), the cognitive domain is also the part of Bloom's taxonomy that is most frequently used by practitioners (see also Anderson & Krathwohl, 2001).

The cognitive domain consists of six levels that are hierarchically dependent on each other. The classification is based on the level of required cognitive processing and on the cognitive complexity. Higher levels represent a deeper understanding of a topic and a higher level of thinking, which in turn represents a higher level of knowledge (Bloom & Krathwohl, 1956). The six levels of the cognitive domain in the original Bloom's taxonomy are illustrated in Figure 3-1.

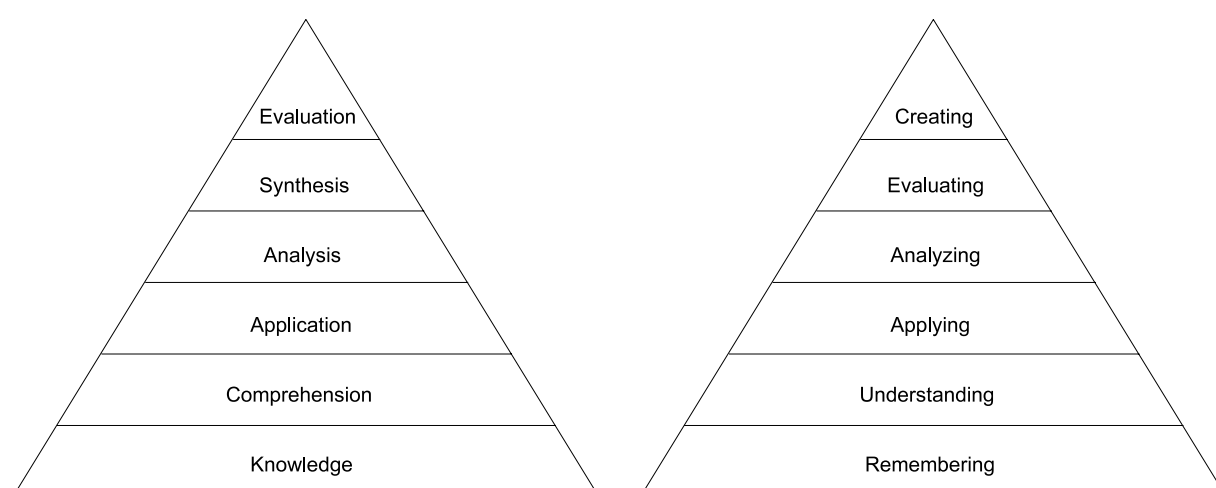


Figure 3-1. The six levels of Bloom's taxonomy. The original taxonomy at the left and the revised at the right. The figure is based on the work of Schult (2007).

In 2001, a revised version of Bloom's taxonomy (Anderson & Krathwohl, 2001) was published. Several researchers who were part of Bloom's original workgroup were involved in the 2001 revision - including Anderson.

The main differences between the old and the new taxonomy are a change of terminology and a change of structure (Forehand, 2005), illustrated in Figure 3-1. These changes reflect some of the changes in pedagogical practice, focus and terminology that have occurred since the first version of Bloom's taxonomy was published. Based on experiences from the use of the old Bloom's taxonomy, the revised Bloom's taxonomy has a stronger emphasis on the applicability as a tool for practitioner's work with instruction and curricula (see, e.g., Anderson & Krathwohl, 2001; Forehand, 2005).

A new knowledge dimension has been added in the revised version of Bloom's taxonomy. The knowledge dimension has four different levels: *factual knowledge*, *conceptual knowledge*, *procedural knowledge* and *meta-cognitive knowledge*, which represent the types of knowledge to be obtained. The different levels of the knowledge dimension are connected to different aspects of the cognitive process dimension through a set of categories and sub-categories (Anderson & Krathwohl, 2001). Together this forms a grid that can be used by practitioners to apply Bloom's taxonomy to their practice. Table 3-2 illustrates this.

The reason for using the revised version of Bloom's Taxonomy is that it better reflects modern learning and teaching practices. The new knowledge dimension in the revised taxonomy can be used to better understand which cognitive activities are involved at different levels of the cognitive process (see Table 3-2). This cognitive mapping can be used to analyze the requirements that must be put on the instructional design, and with that also the requirements for its representation in systems.

The Knowledge Dimension	The Cognitive Process Dimension					
	<i>Remember</i>	<i>Understand</i>	<i>Apply</i>	<i>Analyze</i>	<i>Evaluate</i>	<i>Create</i>
<i>Factual knowledge</i>	List	Summarize	Classify	Order	Rank	Combine
<i>Conceptual Knowledge</i>	Describe	Interpret	Experiment	Explain	Assess	Plan
<i>Procedural Knowledge</i>	Tabulate	Predict	Calculate	Differentiate	Conclude	Compose
<i>Meta-Cognitive Knowledge</i>	Appropriate Use	Execute	Construct	Achieve	Action	Actualize

Table 3-2. The different levels of the knowledge dimension and their connections to the cognitive process dimension. Based on Anderson & Krathwohl (2001, p. 40 pp. 67-68) and Fisher (2007) .

Table 3-2, together with the illustration in Figure 3-1 illustrate the relation between different types of knowledge, the cognitive activities involved and the associated cognitive complexity. These figures may also be compared to the work of Ertmer and Newby (1993) (see Figure 3-2), where instructional strategies associated with different learning theories are compared based on the level of the required cognitive processing and learner's level of task knowledge.

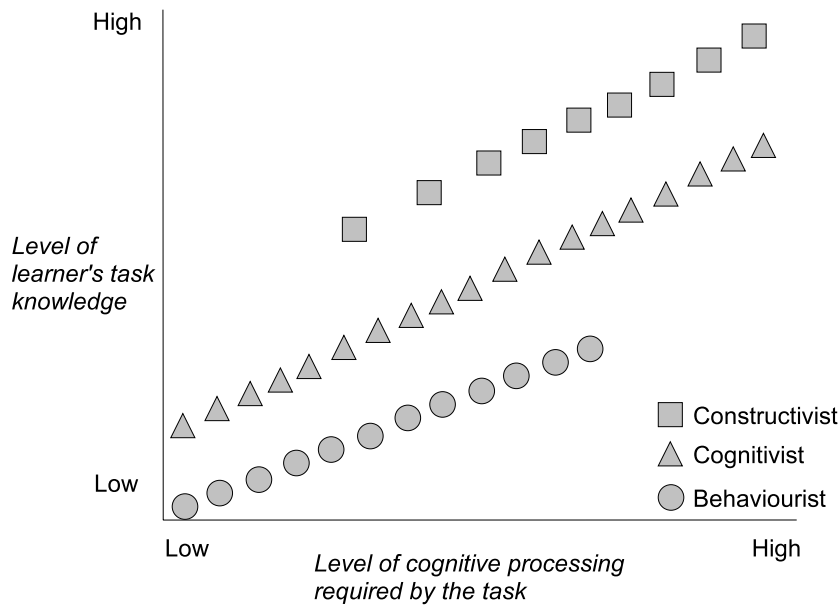


Figure 3-2. A comparison of the associated instructional strategies of the behavioral, cognitive and constructivist viewpoints based on the learner's level of task knowledge and the level of cognitive processing required by the task (Ertmer & Newby, 1993).

3.4 Instructional design and instructional design theory

Apart from learning theories, the development of TEL has been under strong influence of *instructional design* and *instructional design theory*.

There are huge amounts of published work on instructional design and instructional design theory, representing different theoretical viewpoints and different pedagogical approaches. As a research field, instructional design is very broad and full of nuances and directions. This is illustrated by the extensive work presented in several anthologies (see, e.g., Reigeluth, 1983; Reigeluth, 1999; Wiley, 1999; Reigeluth, 2008). This makes it hard to establish a clear and explicit definition of what instructional design is, and what it is not. I have made a selection based on an estimation of how influential the work has been, how often it has been cited, and (obviously) based on the relevance for this thesis.

This chapter is intended as a brief orientation in the field of instructional design theory and instructional design. The objective is to provide a framework for understanding and explaining some of the issues involved when practicing instructional design in connection with TEL and the representation of learning design in learning technology. This understanding is necessary in order to address research question 3. Furthermore, an understanding of instructional design and its practice, in relation to learning theory and the evolvement of TEL, places modularity into context. This is relevant for addressing research question 1 and 2, and for understanding why modularity is important, what kind of modularity is needed, and how modularity should be implemented.

Instructional design and instructional design theory deal with the theories of designing pedagogical instruction and the structuring of learning material. According to Reigeluth (1999c), instructional design theory is actually a collection of theories that has evolved during the 20th century and that deals with methods of instruction. Instructional models, instructional strategies, etc., represent the practice of *applying* the theories to instructional development processes. However, among pedagogues and in everyday speech instructional design theory,

instructional theory, instructional models etc., are often referred to as just *instructional design*. Reigeluth (1999c) also points out that instructional design theory differs from many other theories in that it is design-oriented and probabilistic rather than deterministic - a characteristic that is suitable for the nature of learning.

Reigeluth (1999c) summarizes instructional design theories as:

“... Instructional-design theories are design oriented, they describe methods of instruction and the situations in which those methods should be used, the methods can be broken into simpler component methods, and the methods are probabilistic.” (p. 7)

Merrill (1999) explains instructional theory as being concerned with two primary considerations: *What to teach and how to teach*. This definition is actually similar to some definitions of didactics. Didactics is concerned with the same questions, and sometimes also the question of *why* something should be taught, (see, e.g., Jank & Meyer, 1997).

I will use the term “instructional design” to describe the process of applying instructional design theory to learning through the analysis of learning needs for the purpose of planning and designing pedagogical instructions for implementation with learning technology. A definition that is compatible with Reigeluth’s definition, cited above. I limit the use of the term “instructional design” to the use in connection with ICT.

There are an abundance of instructional design theories that are influenced by, and adapted to, different learning theories. The development of the instructional design research field has followed (or rather interplayed with) the pedagogical developments and trends in general.

A couple of examples of instructional design theories are: Gagnè’s (1977) *Hierarchical Task Analysis* (see also Saettler, 2004, p. 346), which provides a model for sequencing based on eight levels of learning sorted by increasing complexity, CDT (see also chapter 3.2) by Merrill (1983). CDT places the emphasis on learner control. The *Instructional Transaction Theory (ITT)*, also discussed by Merrill, Li, & Jones, (1991) and Merrill (1999), is set out to be an alternative to the early frame-based approaches (Merrill, 1985). ITT is a pattern-based, algorithmic system based on the concept of instructional transactions rather than the traditional sequential and branched approaches of frame-based systems. ET (see also chapter 3.2), described by Reigeluth (1999b), is an attempt to step away from teacher and content-centered teaching, in favor of learner-centered instructions. This shift was accomplished by focusing on guidance in the process of selection and sequencing of content. Several overviews of instructional design theories and methods are available (see, e.g., Reigeluth, 1983; Tennyson, Schott, Seel, & Dijkstra, 1997; Charles M Reigeluth, 1999a; Wiley, 2002b; Merrill, 2002; Reigeluth, 2008).

Merrill (2002) argues that there are a number of principles that are common to all instructional design and he refers to them as *first principles of instruction*. Merrill identifies five common principles: 1) *Learning is promoted when learners are engaged in solving real-world problems*, 2) *learning is promoted when existing knowledge is activated as a foundation for new knowledge*, 3) *learning is promoted when new knowledge is demonstrated to the learner*, 4) *learning is promoted when new knowledge is applied by the learner*; and 5) *learning is promoted when new knowledge is integrated into the learner’s world*. However, many implementations of learning technology fail to implement all of those principles.

Learning Object Design and Sequencing Theory (LODAS) is a theory that was introduced by Wiley (2000). LODAS is of relevance to this thesis as it specifically addresses the instructional use of learning objects. Wiley (2000, pp. 8-11) criticizes the work on learning technology standards (see chapter 3.6), and especially the LOM-standard, for not considering the instructional design aspects of learning objects. Wiley (2000, p. 9) argues that if the

standard bodies do not consider instructional design, neither will teachers. He points out that: “...instructionally grounded sequencing decisions are the heart of the instructionally successful use of learning objects”. Beside the problems that may be caused by not considering the sequencing aspects (i.e., the instructional combination of learning objects) of instructional design, there will be problems regarding the scope of learning objects (i.e., how big should a learning object be). Wiley (2000, p. 11) claims that the granularity is a key factor for learning objects and he points out that this problem is in principal caused by the lack of a general taxonomy for learning objects. Learning objects and learning object taxonomies are discussed in chapter 3.6, where Wiley’s Atomic learning object taxonomy is also discussed. The primary goal of LODAS is “... to provide scope and sequencing guidance for the design of learning objects.” (Wiley, 2000, p. 25).

The LODAS theory is a synthesis of four existing theories: *Elaboration Theory*, *Work Model Synthesis*, *Domain Theory*, and the *Four-Component Instructional Design model*. Wiley claims that LODAS is primarily targeting complex cognitive problem solving and is at the same time compatible with several existing and established instructional theories and is especially adapted for use with learning objects. Wiley argues that the ability to adapt to individual differences have been left out in some of the theories, but that the combinations of instructional design theory and learning objects is a way of addressing individualization issues and adaptability. The adaptation to existing instructional design theories in combination with the focus on learning objects lead to a theory made up of two distinct parts: *instructional design prescriptions* and *learning object prescriptions*. The first part considers instructional design issues and the second part consider how those can be applied to learning objects in creating modules for learning (Wiley, 2000).

The LODAS theory illustrates an attempt to address learning that contains a high level of complex cognitive processing using instructional design and learning objects. As with most instructional design it is dependent on modularity, and the more complex the cognitive processing becomes (see chapter 3.3), the more sophisticated modular models are needed. Merrill’s first principles of instruction (Merrill, 2002) implies that there is a need to support “everything” in order to cover all learning needs and all situations that may occur. The implication of this is that the models and taxonomies for modularity need to be sophisticated and flexible, both from the pedagogical view-point, in order to support all kinds of instructional approaches, and from the technical view-point, in order to represent all possible instructional approaches and all types of learning. Even though the LODAS theory consists of one part for instructional design prescriptions and one part for learning object prescription, it does not consider the technical properties of learning objects. This is an issue that was addressed in paper 1 by suggesting six work areas to establish technical quality criteria that deals with technical properties of learning objects.

As previously indicated, instructional design theory, and especially the pedagogical practice of instructional design is often considered in relation to learning theory (see, e.g., Schunk, 1991; Merrill, 2002; Ertmer & Newby, 1993; Snelbecker, 1999).

3.4.1 Learning theories and implications for instructional design

As illustrated by Figure 3-1 and Figure 3-2, learning theories support some types of learning better than others. The differences are related to the character and level of the cognitive processing involved and to which factors that influence learning. According to Ertmer & Newby (1993), this has implications for the practice of learning design as well as for teaching and learning in general. Different learning theories were compared in Table 3-1, and Ertmer and Newby (1993) compare the effect of different learning theories for the practice of

instructional design. In order to do this they added two questions to Schunk's list (see chapter 3.2.4):

- *What basic assumptions/principles of this theory are relevant to instructional design?*
- *How should instruction be structured to facilitate learning?*

The answers to those questions, based on the study presented by Ertmer & Newby (1993) extend Table 3-1 to include the questions above (see Table 3-3).

The comparison in Table 3-3 is mainly relevant in order to determine the required properties for representing instructional design (or equivalent) in learning technology, addressing research question 3. This is discussed in chapter 5. It also has some relevance for addressing of research questions 1 and 2 as well, in that it indicates the diversity that needs to be covered by a modular approach.

	Behaviorism	Cognitivism	Constructivism
<i>What basic assumptions/principles of this theory are relevant to instructional design?</i>	- Many characteristics and assumptions are embedded in current instructional design practice. - Emphasis on producing observable outcomes in students. - Pre-assessment of students to determine where to begin. - Mastering early steps before more complex, sequencing of instructional presentation. - Reinforcement to affect performance. - Use of cues, shaping and practice to secure stimulus-response.	- Many principles are similar to behaviorism but for different reasons. - Feedback for guiding and supporting accurate mental connections. - Analyze learner to determine predisposition, and learner's mental structures for designing instructions for ready assimilation. - Focus on active involvement of learner and learner control. - Emphasis on structuring, organizing and sequencing information to facilitate optimal processing. - Encourage linking to prior knowledge. - Understanding that students' previous knowledge influences learning outcomes	- Knowledge is linked to the context in which the skills will be learned and applied, as well as to previous experiences. - Focus on assisting learners in actively exploring complex topics to construct their own understanding that is validated through social negotiation. - Learner control, and manipulation of information. - Content is not specified and may be derived from many sources. Not necessary controlled by the teacher. - Presenting information in a variety of ways. - Supporting problem solving that allows learners to go beyond the presented information and content. - Performance goals are related to the process rather than to content. - Assessments focus on transfer of knowledge and skills.
<i>How should instruction be structured to facilitate learning?</i>	- Presentation of the target stimulus and provision of opportunities for the learner to practice giving proper response. - Cues are used to trigger the proper response and for linking stimulus-response pairs. - Arranging the environmental conditions to stimulate the right responses. - Use reinforcement to strengthen the association.	- Make knowledge meaningful and help learners organize and relate new information to existing knowledge in memory. - Instructions must be based on students existing mental structures. - Analogies and metaphors are used. - Arranging practice with feedback to enhance assimilation and accommodation. - Relating new tasks to similar procedures.	- Instruct students on how to construct meaning and model construction of knowledge. - Promote collaboration, design of authentic learning environments. - Cognitive apprenticeship, collaborative learning, cognitive flexibility, social negotiation. - Anchoring learning in meaningful contexts through modeling and coaching. - Learning objectives are not pre-specified and instructions are not pre-designed.

Table 3-3. A summary of the main conclusions of Ertmer and Newby (1993) on the effects of different learning theories on instructional design practice.

Ertmer and Newby (1993) suggest that there is a continuum of learning theory that ranges from behaviorism to constructivism where the focus gradually shifts from teaching to learning. This is a key issue because teaching can be fairly predictable while learning is not. The sequence of instructions can be determined beforehand when using a behaviorist approach, while that is not possible using a constructivist approach (see, e.g., Greeno, 1989). In fact, in the latter case it is often not even possible to predict what content will be used,

which path learners will take, or when and how social interaction and collaboration will occur (See, e.g., Schunk, 2003, pp. 285-328).

Most learning situations address several types of knowledge, leading to tasks that require different levels of cognitive processing, and that span over different dimensions of knowledge and cognitive processing, as illustrated in Table 3-2. As different learning theories support different types of knowledge better than others this means that learning technology must be able to support all approaches at once. This requires the technology to be flexible as well as adaptable from the computer science perspective.

A key factor for achieving flexibility and adaptability is modularity, and a key factor for adaptivity is the ability to represent instructional design and pedagogical process in such way that it can be regenerated based on input from users, preferences or feedback generated by the system (see e.g., Koper, 2004); M. Nilsson, 2001; Devedzic, 2004; Richter, Nejdil, & Allert, 2002). Alteration of the pedagogical representation may also result in that the VLE needs to be altered at run-time.

3.4.2 Instructional design and the effects on TEL

When moving toward constructivism in the learning theory continuum (see Figure 3-2), the learning process becomes personal and learner-centered. This makes it harder to determine the pedagogical process in advance and to outline a generic instructional design that represents learning as a generally applicable process. The situation becomes even more complicated when social interaction and social knowledge building are considered as a part of the pedagogical process. All participants in a learning situation have different background knowledge, different mental models and assimilate knowledge in different ways. This creates a process of joint and mutual “knowledge shaping” that involves learning content, social interaction, testing and rejecting, and consensus building. These activities can be supported using a number of knowledge management techniques combined with different tools for collaboration and social interaction, but it will be virtually impossible to predict how, where, and when the different parts and components of this process will be applied. Consequently, it becomes impossible to describe the pedagogical process as a predetermined sequence of activities and interactions involving digital and human resources. It will only be feasible to design for small parts of the overall pedagogical process, suggesting best practice, while the order of activities, the learning content, other data, and details of how interactions and activities are managed need to be generated at runtime based on feedback from the VLE, from the actors in the system, and from input originating from the deduction of existing and generated data.

The concept of instructional design may appear to be overwhelmingly complicated for many practicing teachers and pedagogues, but in its essence it can actually be compared to a well-structured and well-written lesson or course plan. A lesson plan can be a useful tool for teachers, as well as for students, as long as it is used in a pragmatic way. However, when teachers teach in a classroom and realize that the situation is not the expected or that the situation is changing for some reason, there is a possibility to step away from the lesson plan, which constitutes the instructional design, and improvise. That is, based on input and feedback, the instructional design can be changed at run-time to adapt to the new situation. The new in the situation can, for example, be that students do not react as anticipated, they turn out not to have the expected prior knowledge, or it simply turns out that the learning objectives have to be changed as other objectives have become more important for the students in the current context. The need for changeability may also arise as a consequence of individualization.

This kind of adaptability, based on sensibility and perceptiveness, is one of the things that characterize a good pedagogue. Adaptive and responsive behavior is feasible in a classroom situation, where the instructional design is represented by a lesson plan (that sometimes only exists in the head of the teacher), and the semantics of the instructional design are interpreted and implemented at run-time by the teacher.

The situation is different in a VLE where instructional design is represented using technology that may only support sequencing that is either hard-coded or implemented using an XML-based meta-language, such as *SCORM Simple Sequencing and Navigation* (Capone, Christensen, Falls, & Fletcher, 2004) or *IMS Learning Design* (Koper, 2005). Meta-languages representing instructional design are discussed in chapter 3.10. Such approaches require the pedagogical process to be predetermined in order to be representable in a VLE. This makes pedagogical instruction much more inflexible as it becomes fixed or at least severely limited and cannot be altered at runtime based on feedback from the actors in the system or based on feedback from the system itself. However, the way that pedagogical process and instructions are represented by technology is only one part of the problem. VLEs also need to be able to adapt to changes that are inflicted by changes of pedagogical instructions. This affects the architecture and design of the underlying infrastructures and systems (see discussion in chapter 5). These are issues addressed by research question 1 and 2.

Knowledge about the history and origin of instructional design and TEL may provide an explanation to why pedagogical instruction and TEL are implemented as they are and to why today's learning technology has the characteristics that it has. It may also contribute to the understanding of some of the implications and potential problems of instructional design in relation to its application to learning technology. It should be emphasized that it is not mainly instructional design itself that causes the problems. The main problems are caused by how instructional design is represented in systems, which makes it hard to represent instructional design in VLEs while still maintaining pedagogical neutrality (see discussion in chapter 5). Pedagogical neutrality becomes an issue in order to support learning theories that are not rule-based (algorithmic) and predictive in the way that behaviorism and to some extent also cognitivism are. Pedagogical neutrality and flexibility are qualities that are often emphasized and discussed in the learning object community (see, e.g., Quinn & Hobbs, 2000; Friesen, 2004a; ADL, 2004; Bratina, Hayes, & Blumsack, 2002). However, Friesen (2004b) argues that pedagogical neutrality is hard, or even impossible to achieve with today's technology. Others express similar views (cf. Parrish, 2004; McCormick, 2003).

The development paths and current states of learning technology implementation are discussed in the following chapters.

3.5 Technology enhanced learning

This chapter provides an overview of TEL and its different development paths. The purpose is to emphasize the context within which learning theories, instructional design, and learning technology interplays.

Much of the TEL that we see today still seem to be influenced by the ISD approach that was used by the US military during the Second World War (see, e.g., Gagne, 1962; Saettler, 2004), which continued to evolve and develop after the war. Even though there were several earlier attempts to use technology for learning, such as Pressey's (1927) teaching machine (see also Skinner, 1960), I have chosen to start with instructional system design as it marks a turning point that led on to the branches of TEL that we are still struggling with.

The rationale for the development of ISD (later instructional design) was a need for methods that facilitated fast training of military personnel. The training was first carried out using non-

digital technology, such as instructional films (Saettler, 2004, pp. 184-194). The more complex the weapon systems got, the more important appropriate training became. Learning theories originating from behaviorism (see chapter 3.2.1) were appealing, as well as suitable, for the learning needs of the army. Most of the skills needed by army personnel were of a typical drill and practice nature, composed out of a series of predestined, relatively simple and straightforward tasks that could be sequenced together to form larger units of training. An example of how this was implemented in its early days was the machines for individualized self-instruction called “phase checks” (see, e.g., Saettler, 2004). English (1942) invented a similar device already in 1918. The approach of decomposing learning into smaller modules that are later sequenced together (also referred to as *chunking and sequencing*) is in many ways still representative for how learning and the delivery of digital learning content is often implemented with technology, one such example is SCORM (Capone, et al., 2004).

3.5.1 Programmed instruction and programmed learning

The military needs for education were still the driving force that gave stimulus to the developments of instructional design and learning technology long after the war (see, e.g., Saettler, 2004; Snelbecker, 1999). However, ideas and methods from instructional design were slowly transferred to education and training outside the military sector and instructional design started to spread to other parts of society in the fifties (see, e.g., Snelbecker, 1999). This development was still led by behaviorists such as Skinner, Gagne and others (see, e.g., Saettler, 2004, pp. 394-401; Snelbecker, 1999). Eventually, this led to the birth of *programmed instruction*, which was implemented by the teaching machine (see, e.g., Skinner, 1958; Homme, 1960; Skinner, 1960; Meyer, 1960). Programmed instruction was based on a modular approach where training materials were decomposed into small modules referred to as frames. The frames were then sequenced together in a linear fashion where all learners are obligated to follow the same linear path. Later implementations of programmed instructions used a slightly more sophisticated, intrinsic (branched) approach supported by simple interactivity. The consequence was that several different paths could be programmed and learners can be directed to different paths depending on input, such as prior responses (see, e.g., Norman A Crowder, 1960; Pask, 1960). This creates a higher degree of interactivity. Crowder (1964) discusses the differences between linear approaches and intrinsic approaches in detail.

Programmed instruction emphasized self-teaching by providing atomized, self-contained and self-paced training using technology (see, e.g., Homme, 1960) even though Lysaught and Williams (1963) also emphasized the important role of teachers. The basic principle of programmed instruction is, in general terms, to walk through instructions in an incremental way where learners progress is successively tested and proper responses are reinforced (see, e.g., Orlich, 2000).

Programmed instruction later evolved into *programmed learning*, which could be regarded as a variation (or a specialization) of programmed instruction. The evolvement of programmed learning came out of a desire to implement a model that was better adapted for use in schools. The core ideas of programmed learning were in most respects similar to the ideas of programmed instruction, but one of the differences between programmed instruction and programmed learning was that programmed learning had a more expressed focus on the user. User characteristics, such as prior knowledge, were considered while programmed instruction focused on modifying behavior. This development seems to mark a move toward cognitivistic views of learning. A typical programmed learning application would be learning material consisting of small chunks, still referred to as frames, and the pedagogical activities were usually of the “fill in the blanks” types. Learning was reinforced through immediate feedback

on learner activity (e.g., Saettler, 2004). The distinction between programmed instruction and programmed learning is often not made and the terms are often used as synonymous. The differences are often unclear and ambiguous, and the ideal application of both is the automatic teaching machine.

3.5.2 Computer-assisted instruction and computer-based training

Eventually, starting in the sixties with projects such as the PLATO-project (Bitzer, Lyman, & Easley, 1965), programmed instruction and programmed learning evolved into *computer-assisted instruction (CAI)*, also referred to as *computer-based training (CBT)* or *Computer-based Learning (CBL)*. This development appears to have been triggered by the adoption of more advanced instructional theories that were beneficial for instructional design, such as Gagne's (1977) nine events of instruction that provided a tool for designing better instructions, and Bloom's Taxonomy (Bloom & Krathwohl, 1956) (see chapter 3.3.1) that provided a tool for defining and categorizing educational objectives. However, according to Saettler (2004, pp. 307-311), most of the early CAI implementations arose from the behaviorist tradition, typically of drill-and-practice type, while CAI gradually slid toward cognitive learning theory during the seventies.

The shift from programmed instruction to CAI also indicates a shift from mechanical learning machines to modern computer-based approaches that were more similar to the kind of TEL used today. Compared to what was commonly the case within programmed instruction, CAI made better use of computer power, and used more sophisticated, intrinsic systems, for organizing and managing content, utilizing technologies such as branching, interlinking and sequencing. While still emphasizing characteristics like self-paced and self-contained instructions that are managed and delivered using computers, CAI emphasized specialized training software and interactivity. CAI incorporated a higher number of different types of learning-related activities, such as diagnostic testing, game-like training, quizzes, student monitoring and likewise, and was usually considered to be more efficient than previous approaches (see, e.g., Fletcher-Flinn & Gravatt, 1995; Johnson, 1985).

During the nineties there was a shift from specialized educational software, as the carrier of CAI, to CD-ROM and multimedia. Focus was turned toward the delivery of interactive multimedia learning content. A significant difference was that the development process changed and multimedia experts, rather than software developers, were producing the learning content. Multimedia content was also less expensive to produce – even though it was still very expensive. It also became possible for skilled pedagogues to develop their own learning material using relatively uncomplicated multimedia development software, such as Multimedia Lab and HyperStudio (see, e.g., Havice & Galloway, 1998; Haviland & McCall, 1999). The breakthrough of multimedia on CD-ROM also meant the breakthrough of digital media, such as digital images, digital audio and digital video. Audio and video have traditionally been used in education (e.g., Saettler, 2004, pp. 123-129), and the introduction of digital media in combination with interactivity created new pedagogical opportunities – as well as challenges. Many of the concepts from the multimedia and CD-ROM era were transferred to the web where it created new pedagogical and technical challenges that were bound to occur in a networked environment (see chapter 3.8.3).

The term (as well as the idea of CAI) is still in use, even if the focus is gradually changing. Generalized VLEs are taking over more and more of the tasks of CAI, and specialized training software is becoming generalized and incorporated into VLEs and learning content.

Learning content of the type that was previously delivered as specialized training software is now being delivered as learning objects, intended for use in VLEs and especially in LMSes¹⁷.

One of the major drawbacks of specialized training software is that it is (and always has been) expensive to develop, which makes it hard to gain profitability in the production of digital learning content. This is also an important reason for the emphasis on reusability within the learning object community. Learning objects are further discussed in chapter 3.6.2.

3.5.3 Web-Based Training and Web-Based Learning

In the mid-nineties, the breakthrough of the web led to an evolvement that took TEL to the Internet in the shape of *Web-Based Training (WBT)*, *Web-Based Instruction (WBI)*, *Web-Based Learning (WBL)*, and similar concepts. I will use the term Web-Based Learning, for similar reasons that I use the term TEL (see chapter 1.7.2).

The move from CAI (and similar approaches) to web-based approaches is likely to be one of the most important occurrences in the evolution of TEL. Most significant in this development is that the “Computer” was exchanged with the “Web” as the mediator of education and learning, which indicates learning is now delivered via the Internet and the web. That is, TEL in all its forms moved to the Internet. This transition marks a change in view from the local perspective, with the personal computer in focus, to a global view with the Internet and the web in focus. The breakthrough of WBL occurred in the latter half of the nineties and the use of the Internet and the web as delivery channel has since then placed the focus on collaboration and social interaction rather than on individual training, which in turn has created new challenges for VLEs and learning technology in general (see, e.g., Piccoli, Ahmad, & Ives, 2001). At the same time WBL is still struggling with some of the limitations of web technology, which was not an issue before the web. A paradox of using the web for TEL is that it introduces endless possibilities in terms of independence of time and place, social interaction, delivery of content and services etc., but at the same time it introduces some rather severe technical limitations that prevent TEL to actually gain full advantage of all, in many respects still theoretical, possibilities (see chapter 3.8.3).

The evolvement of TEL and learning technology is illustrated in Figure 3-3. The figure also illustrates how things are interconnected and how the technological development and the pedagogical development are (from a TEL point of view) communicating vessels.

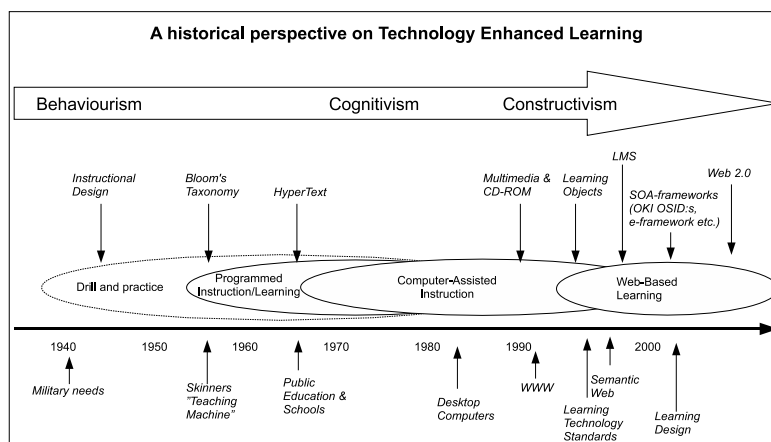


Figure 3-3. The evolvement of TEL and learning technology from a historical viewpoint.

¹⁷ Because there are many views of what is and what is not considered to be an LMS respectively a VLE I use the term LMS as specialization or a sub-type (or branch) of VLEs.

3.6 Modularization of TEL

The interrelation between technology and learning, illustrated in Figure 3-3, is likely to be one of the driving forces behind some of the recent trends within TEL. Some of those trends are discussed from a modularity point of view in this chapter.

3.6.1 Learning Technology Standards and modularity

The increased awareness of the importance of standards has led to similar processes for developing standards within the learning technology domain as within most other domains with special needs. One of the cornerstones in the development of domain-specific standards is the utilization of existing general technology standards as the basis for domain-specific standards (Duval, 2004). While the Internet has changed system borders, and in some cases even removed them, it is the use of open standards that has made it possible.

TEL benefits from this development as learning and TEL-services can flow more independently of organizational and geographical borders and vendor lock-ins. One example is the rapidly increasing amount of free and globally available digital learning content¹⁸ from prestigious and well-known educational institutions, a phenomenon that is analyzed and discussed by Wiley (2007) and in OECD (2007).

Many organizations have started to realize that the management of information and ICT-systems is no longer only a matter for their own organization, it is rather a matter of being a part of a global infrastructure where information and services are raw material that can be used as building blocks for the creation and delivery of new services and refined information (see, e.g., O'Reilly, 2006). The ability to share information, (or pieces of information) and data through communication with other systems and services has become crucial for many organizations. This is also one of the driving forces behind the so-called Web 2.0 movement. However, Web 2.0 is an ambiguous term that has no commonly accepted definition (cf. Anderson, 2006; Shaw, 2005; O'Reilly, 2006). In spite of this, the technologies and philosophy that are usually associated with Web 2.0, such as social and collaborative software, have already become essential for education (see, e.g., Salmenjoki, Paulsson, & Vainio, 2007; Avram, 2006; Anderson, 2007).

The importance of open, international standards as a crucial component for modern infrastructure in a global context is becoming more and more recognized and several organizations are focusing on the development of standards for learning technology, which I will refer to as *learning technology standards*. Among the most influential standard organizations are the *IMS Global Learning Consortium (IMS)*¹⁹, *IEEE*²⁰ and their *LTSC* initiative²¹, *Dublin Core (DC)*²² for metadata, and the *ISO-ITLET* initiative (*JTC-1 SC36*²³). Most organizations, and the specifications they produce, have cross relations that are discussed by Duval (2004).

One of the most influential organizations on the use of learning technology standards is the *Advanced Distributed Learning (ADL)*²⁴ (Masie, 2003). ADL is the maintainer of the

¹⁸ Often referred to as Open Educational Resources (OER) or Open Courseware.

¹⁹ <http://www.imsproject.org/>

²⁰ <http://www.ieee.org/portal/site>

²¹ <http://ieeeltsc.org/>

²² <http://www.dublincore.org/>

²³ Standards for: Information Technology for Learning, Education and Training, <http://www.jtc1sc36.org/>

²⁴ <http://www.adlnet.gov/>

Shareable Content Object Reference Model (SCORM) (ADL, 2004). SCORM is a reference model that prescribes how to implement standard specifications that are mainly developed and promoted by other organizations, such as the IEEE and IMS. In addition, SCORM implements its own application profiles of some specifications, such as metadata, and sequencing and navigation, as well as some SCORM-specific technologies such as the *SCORM Run-Time Environment (RTE)*. SCORM is formally not a de jure standard, even if it is often referred to as a standard (Masie, 2003). SCORM is important because it has a great impact on the use of learning technology standards and on the way that TEL is delivered as it influences how VLEs (especially LMSes) are implemented (See, e.g., Downes, 2003; Masie, 2003). The impact of SCORM has given it the status of de facto standard, resulting in that SCORM has been implemented in many LMSes and is widely used for the packaging and delivery of learning objects and training courses.

The main part of the standardization work has been focused on digital learning content, and especially on standards for descriptive information (i.e., metadata), sequencing and navigation, and packaging of digital learning content. Much of the standardization work evolves around modular digital learning content, i.e., learning objects. The focus is slowly changing as the content-centered standards are maturing, and VLEs, system architecture and learning technology infrastructure in general are gaining more attention. Learner-related data is another area that is getting more attention, and where standard specifications are being developed, such as IMS LIP, which is an information model for learner-related data (Wilson & Jones, 2005; IMS, 2005b).

Similar to learning objects, there are a couple of initiatives addressing infrastructure and system architecture that have a certain focus on modularity. Most notable are the work by IMS on *General Web Services (IMS GWS)* (IMS, 2005a), and IMS' recent adoption of the MIT's *Open Knowledge Initiatives (OKI)* and the *Open Service Interface Definitions (OSIDs)* (Thorne & Kahn, 2003), as well as the *e-Framework*²⁵ initiative (Olivier, Roberts, & Blinco, 2005), which was originally initiated by the British *Joint Information Systems Committee (JISC)*²⁶, and has now turned into an international project involving several countries and organizations. The OKI OSIDs are implemented in the *Sakai* project²⁷ (Counterman et al., 2005). The emerging learning architecture models and frameworks also reflect the importance of learner-related information, such as in the IMS GWS, as well as in the e-Framework.

Both the OKI OSIDs and the e-Framework address SOA for TEL, but approach it somewhat differently. The main difference between the e-Framework and the OKI OSIDs is that the OKI OSIDs are Java-centric, implementing a service architecture based on Java RMI, while the e-Framework uses web service technology. Another important difference is the abstraction level of the service descriptions. The e-Framework is targeting specific and detailed pedagogical services, while the OKI OSIDs specify more general-purpose services, and are less specific on pedagogical services. However, this may change over time. Norton (2004) compares the OKI OSIDs and the e-Framework approaches. Norton's comparison gives a good overview of the two frameworks, even though the comparison is three years old and some things have changed, the basic principles and differences remain.

Reference models and frameworks for VLEs, like the e-Framework and the OKI OSIDs, are of relevance for this thesis as they facilitate a breakdown of functions and processes within, as well as outside, the VLE, which creates the basic conditions for a modularization of the infrastructure and system architecture. Even though these initiatives are still in their infancy,

²⁵ <http://www.elframework.org/>

²⁶ <http://www.jisc.ac.uk/>

²⁷ <http://www.sakaiproject.org/>

SOA combined with abstract models for modularity, create the potential conditions for modularity and flexibility for VLEs – much like the learning object concept does for digital learning content. By modeling processes and functionality, using established design principles and paradigms, it becomes possible to determine a framework that breaks down and isolates services and processes, and their associated functionality can be defined and described (Erl, 2007, pp. 26-30). In papers 2, 3 and 4, a modular taxonomy and a SOA framework were suggested and implemented when addressing research question 1 and 2. These results are described and discussed in chapter 4 and 5. Service-orientation is discussed in chapter 3.11.

3.6.2 Learning objects

Modularization of learning processes and learning content has actually been around for a long time. Aristotle (Klinger, 1999) suggested that learning could be subdivided into small chunks and then linked together in hierarchal structures, i.e., chunking and sequencing. He suggested a systemic approach using a hierarchical tree structure that could be used to describe, organize and visualize knowledge through classification. Those are ideas that have been present throughout the evolution of instructional design (see chapter 3.4), and similar models for structure, based on hierarchal classification systems, using a root-node and branches, is still used by some instructional approaches. This idea is also similar to the idea that constitutes the basis for representing information structures using techniques like XML. Modularity, and especially content modularity, has been important during the whole evolution of TEL from the first attempts with programmed instruction and teaching machines until today's web-based approaches with LMSes and learning objects.

Learning objects represent a modular approach for digital learning content. It is also the approach most discussed and, according to some, one of the most promising approaches for achieving a higher level of pedagogical and technical flexibility, and much of the rhetoric that surrounds the learning object phenomenon concerns the pedagogical freedom of teachers and students, a topic that is frequently debated and discussed within the research community (cf. Weller, Pegler, & Mason, 2003; Christiansen & Anderson, 2004; Schluep, Bettino, & Guttormsen Schär, 2005; McCormick, 2003; Rehak & Mason, 2003; Gibbons & Nelson, 2002). Learning objects are often seen as a path to adaptability, adaptivity, reusability, scalability, and instructional independence (see, e.g., Gibbons & Nelson, 2002; ADL, 2001). However, this view is also questioned and some aspects of the learning object phenomenon is criticized. The main reason for this is that learning objects, to a great extent, have failed to live up to the expectations (cf. Dodani, 2002; Friesen, 2004b; Parrish, 2004; Wiley, 2003; McCormick, 2003).

The learning object concept has evolved during the last decade and a lot of research has been carried out on learning objects and related areas. Big issues concern learning object's relation to instructional design (see, e.g., Wiley, 2002b; Wiley, 2000; Bannan-Ritland, Dabbagh, & Murphy, 2002; Rehak & Mason, 2003; Douglas, 2001), and learning object definitions and taxonomies (see, e.g., Wiley, 1999; Gibbons & Nelson, 2002; Quinn & Hobbs, 2000; Song & Andersson, 1999; Wiley, 2002b).

The term learning object occurred in the mid-nineties and was coined by the *Learning Architecture and Learning Object task force (LALO)* within the *American organization Computer Education Management Association (CEdMA)*. The underlying ideas of learning objects were inspired by the object-oriented software engineering approach within computer science. Parts of the TEL community started to realize the potential benefits of a modular approach to digital learning content, similar to the Object-Oriented approach that has become a de facto standard within the software development community. This led to the formation of

the LALO taskforce (Hodgins, 2001). The objectives of the LALO Task Force were expressed by CEEdMA as:

“The vision of the LALO task force has been to enable new and existing learning content to be created as independent Learning Objects, such that they can be assembled in any combination to meet an individual’s learning needs, resulting in increased personal productivity.” (CEEdMA, 1995)

The key enabler to accomplish this was to facilitate the development and use of learning technology standards (see, e.g., Masie, 2003; Duval, 2004; Hodgins, 2000).

The definitions and the meanings of the term learning object has changed over time. Changes have occurred when standards have matured, when implementations have shown that some things did not work as anticipated, and depending on differences in focus or theoretical perspectives. Several learning object definitions exist alongside each other, some definitions are compatible with each other and some definitions are not.

IEEE made one of the first and most basic definitions of a learning object. According to IEEE a learning object is *any entity, digital or non-digital, which can be used, re-used or referenced during technology-supported learning*. IEEE exemplifies learning objects as; *including multimedia content, instructional content, learning objectives, instructional software and software tools, and persons, organizations or events referenced during technology-supported learning* (IEEE, 2002). The IEEE definition is very wide and can be interpreted to include just about anything, which makes the definition somewhat hard to use, and it does not actually say anything about the properties of learning objects. The IEEE definition, which was originally adopted from the CEEdMA LALO working group, has served, however, as a basis for other refined definitions.

Sosteric and Hesemeier (2002) define learning objects as *digital objects that have a formal educational purpose within a predestined pedagogical context*. Sosteric and Hesemeier (2002) take on a rather traditional view on learning, and there is a risk that such a view may limit the pedagogical choices as well as the innovative aspects of using learning objects. Like the IEEE definition, it does not say anything about technical properties. McGreal (2004) claims that the definition should be limited down to units that practitioners already prefer to work with and suggests a definition where Learning Objects are *“...any reusable digital resource that is encapsulated in a lesson or assemblage of lessons grouped in units, modules, courses and even programmes”*. Song and Andersson (1999) approach Learning Objects slightly differently. They argue that Learning Objects should be regarded as decomposable, and that there must be a separation between data, operations and the carrier of the data. They also claim that an object should be described using a set of attributes and relationships to other objects. Song and Andersson focus mainly on the internal structure of Learning Objects and their relations to other objects. The Song and Andersson approach rests heavily on experience and concepts from OOP. A problem in the learning object community is that there are about as many definitions of learning objects as there are people working with learning objects.

Learning objects have often been compared to Lego building blocks. Wiley (1999) criticizes this comparison for being oversimplified. Wiley (2002a) suggests that learning objects are more like atoms, and argues that a learning object cannot be combined with any other learning object, and that proper composition of learning modules requires pedagogical knowledge, subject knowledge, as well as knowledge about instructional design (cf. Friesen, 2004a).

Wiley (2002a) argues that learning objects are like atoms in that they attract some learning objects, while repelling other. According to the atomic model, the rules for how learning

objects can be assembled and used are stricter and their interfaces tend to be more complex. However, it is not important whether learning objects are compared to Lego or atoms; the important characteristic is that the totality (the module) of the assembled learning objects is more than the sum of its parts. Hence, an added value is created when learning objects are assembled and contextualized.

Wiley (2002a) suggests a taxonomy for learning object-types that complies with his atomic view of learning objects. The taxonomy focuses on the characteristics of learning objects, and mainly addresses the type of elements contained, the type of logics contained, and the complexity of learning objects. The taxonomy makes sense from a pedagogical point of view where learning objects are regarded as small pieces of digital (potentially interactive multimedia) content for learning purposes. Wiley's taxonomy is however too abstract and high-level to be really useful from an object and component technology or technical implementation point of view. The main problem is that (like atoms) learning objects do not only need to attract or repel each other pedagogically – they need to interface as well. Wiley's taxonomy is described in more detail below because it is widely referred to, as well as used as a comparison in papers 1 and 2.

Wiley's taxonomy sorts learning objects into five categories (see Figure 3-4), ranging from the most simple *Fundamental learning objects* (e.g., pictures, video, and texts), to the most complex, *Generative-Instructional Learning Objects*. Generative-Presentation/Generative-Instructional learning objects can also include web pages, which could in some cases also be argued to be *Combined-Open Learning Objects*. This is a problem because they can hardly be treated as components from a component perspective (see chapter 3.11) and they are also likely to be inflicted with a high-level of pedagogical context dependence. Fundamental learning objects usually have a high level of context independence, pedagogically as well as technically, and can be regarded as *data objects* that can be used as building blocks for composing composite units (i.e., modules). Technically, these are not limited to learning, but are described using metadata that loosely place them in a learning context. Fundamental learning objects are fairly uncomplicated to handle as they are mainly used as raw building blocks, as in cases where they are non-decomposable media objects, such as JPEG images or MP3 audio or similar.

Learning object characteristic	Fundamental learning object	Combined-closed learning object	Combined-open learning object	Generative-presentation learning object	Generative-instructional learning object
<i>Number of elements combined</i>	One	Few	Many	Few-Many	Few-Many
<i>Type of elements contained</i>	Single	Single Combined-Closed	All	Single, Combined-Closed	Single, Combined-Closed, Generative-Presentation
<i>Reusable component objects</i>	(Not applicable)	No	Yes	Yes/No	Yes/No
<i>Common functions</i>	Exhibit, display	Pre-designed instruction or practice	Pre-designed instruction and/or practice	Exhibit, display	Computer-generated instruction and/or practice
<i>Extra-object dependence</i>	No	No	Yes	Yes/No	Yes
<i>Type of logic contained in object</i>	(Not applicable)	None, or answer sheet-based item scoring	None, or domain-specific instructional & assessment strategies	Domain-specific presentation strategies	Domain-independent presentation, instructional, & assessment strategies
<i>Potential for inter-contextual reuse</i>	High	Medium	Low	Medium	High
<i>Potential for intra-contextual reuse</i>	Low	Low	Medium	High	High

Figure 3-4. Wiley's atomic learning object taxonomy (Wiley, 2002a, p. 17).

Generative-Presentation/Generative-Instructional Learning Objects on the other hand are much more complex to handle. Partly because they are a result of someone's authoring efforts where several elements are aggregated into a composite module – often treated as an instructional unit based on an intended pedagogical use and a specific pedagogical approach including a pedagogical design and accompanying sequencing.

Taken all together, the definitions discussed above show that the common underlying idea of learning objects (irrespective of definition) is to provide a modular approach that facilitates the reuse and interoperability of digital learning content. In theory, learning objects are small and reasonably context-independent (technically as well as pedagogically), platform-neutral, and reusable chunks of content that can be aggregated and disaggregated in order to meet the requirements of a specific learning context. Learning objects produced by different producers are intended to work together, and the overall pedagogical context is created by the way in which learning objects are assembled, and by the way learning objects relate to other learning objects or learning modules. The pedagogical context is supposed to be created using instructional design and represented using some sequencing technology, such as *IMS Simple Sequencing* (IMS, 2003b) or *SCORM Sequencing and Navigation* (Capone et al., 2004). There is, however, an ongoing development toward more sophisticated and fine-grained approaches implemented using *educational meta-languages* that are sometimes referred to as *Educational Modeling Languages (EML)* (Rawlings et al., 2002). EMLs are discussed in chapter 3.10.

As discussed above, an important objective of learning objects is to establish the conditions needed for a high level of flexibility in terms of the composition (as well as decomposition), delivery, use, and reuse of digital learning content. There is, however, some serious issues associated with the ways in which learning objects are implemented – issues that prohibit learning objects to be fully functional (see, e.g., Wiley, 2002a; Rehak & Mason, 2003; McCormick, 2003; Douglas, 2001; Gibbons & Nelson, 2002; Friesen, 2004b; McGreal, 2004). It becomes apparent, when studying different learning object definitions that a common problem is that most learning object definitions and taxonomies express various properties (pedagogical and functional), but say nothing about how those properties should be obtained, or implemented in terms of software architecture, standards or APIs for interacting with other learning objects and with the VLE. Such interactions are essential for realizing some of the properties that are commonly assigned to learning objects. This makes learning objects lose contact with the original source of inspiration: the ideas inspired by OOP, and component technology within computer science (see, e.g., Song & Andersson, 1999; Friesen, 2004b; Schluep, Bettino, & Guttormsen Schär, 2005; Douglas, 2001; Sosteric & Hesemeier, 2002). When disregarding those properties and losing this connection, the concept also loses some of its meaning as it becomes hard (if not impossible) to actually implement many of the properties in a way that is generic enough to fulfill the characteristics that are given to learning objects in terms of reusability, modularity and platform neutrality.

Especially problematic are the properties that are related to the idea of learning modules as context-independent pieces of learning content that can be combined, reused and distributed as “integrated” packages, composed out of the small building blocks that learning objects are intended to be. I will return to this discussion in chapter 5.

3.7 Virtual Learning Environments

As discussed in chapter 1.7, the term VLE is used in a broad and all-encompassing sense in this thesis. There are some specialized branches of virtual learning environments that have had a significant impact on how TEL is being implemented and used, and one of the most influential concepts from that perspective is LMSes.

3.7.1 Learning Management Systems

LMSes started to occur in the late nineties and were a direct result of the increasing application of learning objects, together with the development of learning technology standards.

LMSes were an attempt to put learning objects and learning technology standards into a context that was clearer and better suited for the requirements of trade and industry. This was also likely to be one of the main reasons for the focus on training and performance rather than on learning and education. In this respect LMSes are a natural consequence of the evolution of TEL and instructional design that was described in chapter 3.4 and chapter 3.5. The wide adoption of the LMS-concept has given SCORM (see, e.g., Masie, 2003; Duval, 2004) a unique position as a de facto standard. The reason for this is most likely that SCORM succeeded in presenting an all-encompassing, standard-based framework that was ready to be implemented and used together with learning objects and LMSes. However, SCORM is criticized for contributing to the preservation of some of the aggravating legacy of TEL by being too strict and limited, at the same time it is not regarded as suitable for all learning situations, accused of not being pedagogically neutral and focusing mainly on training and performance (see, e.g., Kraan & Wilson, 2003; Downes, 2002; Friesen, 2004b). Even though SCORM has evolved and become more flexible in later versions, many of the basic problems still remain. Examples of such problems are the way instructional design is represented using *SCORM Sequencing and Navigation* (Capone et al., 2004) and the way learning objects (or *Sharable Content Objects (SCO)* in SCORM lingo) is defined and implemented. See chapter 3.10 and 5 for an in-depth discussion.

There are other terms that are used more or less synonymous with the term LMS. One example is *Learning Content Management Systems (LCMS)*. The term LMS is sometimes even used synonymously with the term VLE.

As with most other terms related to TEL there are several definitions. Paulsen defines LMSes in the following way:

“Learning Management System is a broad term used for a wide range of systems that organize and provide access to online learning services for students, teachers, and administrators. These services usually include access control, provision of learning content, communication tools, and organization of user groups.” (Paulsen, 2002)

Paulsen’s definition is frequently used and has been adopted by MIT OKI, as well as by several other organizations. Several other definitions exist that are similar to Paulsen’s definition. Another example is the definition by Masie:

“Software that automates the administration of training events and contains organizational information such as role-based curricula, learner profiles training histories, competencies, and training resources. A learner’s development plan and job-related training can be stored and personalized to the individual. A LMS provides an environment for learner registration populates courses in a catalogue, records data from learners; and supplies reports to management. A LMS is typically designed to handle courses by multiple publishers and content providers, both internal and external to an

organization. Many LMSs are designed to conform to specifications for content interoperability.” (Maise, 2003)

I will use the broader term VLE throughout this thesis, for the reasons discussed in chapter 1, with the exception of when LMSes are explicitly addressed as a concept. It is important to emphasize that I do not regard the term LMS to be synonymous to the term VLE. *I regard LMSes to be a subtype of a VLE.*

The definitions above indicate that LMSes have a certain focus on the administration and management of learning, courses and learning objects, rather than on pedagogical activity (see, e.g., Ismail, 2002). The pedagogical context and pedagogical design is often provided as a part of the learning content, determined by the way in which the content is sequenced and packaged. This contradicts some of the asserted basic properties of learning objects (See, e.g., McCormick, 2003).

LMSes are criticized for being implemented as so-called information silos (see, e.g., Downes, 2002), which means that the system does not communicate and function very well with the surrounding world. Most LMSes are implemented from a local, isolated perspective rather than from a global learning perspective. The effect of this is that the modular concept of learning objects, intended for sharing, reuse and flexibility, is locked into a silo. This is also the immediate effect for all other data that is managed and used within the VLE, such as learner information and performance data. This effect is created by vendor lock-ins and monolithic thinking, as well as by too restricted and unsophisticated standards (see, e.g., (Downes, 2002; M. Nilsson, Johnston, Naeve, & Powell, 2007; M. Nilsson, 2001). Those issues are discussed in detail in chapter 5.

3.8 Enabling technologies

3.8.1 Hypertext technologies

If one single technological development would be chosen that has had an unequalled impact on TEL, hypertext would probably be a very good candidate. Hypertext had its major breakthrough, in the early nineties, as the technology upon which the web was built. However, the hypertext history is older than that. The first steps were taken by Bush (1945), who presented the idea of the Memex (memory extender). The term hypertext did not turn up until 1965 when Nelson (1965) used it when he presented a hypertext-based *Evolutionary File Structure (ELF)*.

In the eighties hypertext-based systems and applications started to become publicly available. Xerox released the Lisp-based NoteCards system in 1985 (Monty & Moran, 1986; Trigg & Irish, 1987). The NoteCards system was unique in the sense that it was window-based and unusually user-friendly. Another early hypertext application that became one of the most used applications for personal computers (not only because it was bundled with all new Macs) was Apple's *HyperCard*, introduced in 1987. HyperCard was implemented using an index-card metaphor where index-cards, organized in "stacks", contained the data and application logics that were programmed using a built-in programming language called *HyperTalk*. The information was interlinked using hyperlinks that related information, cards and stacks to each other. Links were represented using buttons, forms and other elements for interaction. This was a practice that was criticized by Hall (1994) for reducing the dynamic abilities of hypertext by hard-coding links instead of dynamic generation of hyperlinks. Although HyperCard was presented using the metaphor of an index-card system, it was used to develop digital learning content as well as full-fledged applications.

The first major hypertext conference (Hypertext'87) was held in 1987. The major breakthrough of hypertext came in 1991 when the web was introduced as a global, distributed hypertext network built on top of the already existing Internet topology (See, Nielsen, 1995).

3.8.2 Standard Generalized Markup Language (SGML)

The underlying ideas of hypertext can be implemented in several ways, and one of the more important inventions for the wellbeing of hypertext was the SGML, introduced in the seventies. SGML is a meta-language that can be used to define new markup languages. A markup language that is defined using SGML does not have to be a hypertext markup language, but defining hypertext markup languages has become one of the most important uses of SGML. SGML was based on the work with the *Generalized Markup Language (GML)* carried out in the late sixties (Goldfarb, 1996).

Applications of SGML are usually implemented using *Document Type Definitions (DTD)* or *schemas*, which are basically a declaration of well-formed syntax for a specific markup language. One of the major ideas of SGML (and later XML) is the concept of separating structural markup and syntax from formatting and presentation markup. However, it is still possible to define markup languages that do not follow this rule. The original markup language of the web (html) is an example of this malpractice. This is something that has created problems ever since the birth of the web, as the data part of html is not machine-readable in a way that makes it possible to process data separated from presentation and structure. The main reason for this is that tags for defining structure and formatting instructions are mixed with data. This means that there is no good way of knowing what is what, and there is no well-formed syntax or semantics that can be easily extracted from the data parts of an html-file.

While html is likely to be the most well known application of SGML (Nielsen, 1995), there are also other SGML applications. One example is the *Text Encoding Initiative (TEI)*, which is targeting the markup of literary and linguistic texts (see, e.g., Ide & Sperberg-McQueen, 1995). Other examples of markup languages implemented as applications of SGML are *DocBook*²⁸ for technical documentation, and *HyTime* for multimedia (e.g., Goldfarb, 1991).

Since its birth, SGML has been used as the basis for defining complex markup languages, often in highly specialized contexts, such as for technical documentation. Such markup often includes both structural markup and semantic markup. There are, however, several important issues that need to be dealt with when mixing semantics with structural markup, especially in a distributed, eco-system-like environment such as the web (see, e.g., M Nilsson, Palmér, & Naeve, 2002). A good practice is therefore to separate semantics from syntax and from presentation, which is one important driving force of the Semantic Web (Berners-Lee, Hendler, & Lassila, 2001) (see also chapter 3.9.2).

SGML is also the origin of its better-known relative: the *Extensible Markup Language (XML)*, which, like SGML, is a general-purpose markup language. SGML has been criticized for its complexity, and this was one of the main reasons for the development of XML, which can be regarded to be a simplification of SGML (e.g., J. Clark, 1997; Bray, Paoli, & Sperberg-McQueen, 1998). XML is mainly intended to be used for describing the structure and syntax of data and is commonly used for purposes such as encoding of documents, for data exchange and for serializing data. XML consists of a family of related technologies used for different purposes, such as XPointers, xPath, which are used for addressing in XML documents, and the *Extensible Stylesheet Language Transformation (XSLT)* that is used for transformation of XML into other XML based documents or other document formats (see, e.g., Bray, Paoli, & Sperberg-McQueen, 1998; J. Clark & DeRose, 1999; Nussbaumer & Mistelbacher, 2002). During its short lifespan, and because of its relative simplicity, XML has become a multipurpose defacto standard for exchange of data and for encoding documents.

3.8.3 The Internet and the World Wide Web

The breakthrough of hypertext had a great impact on how learning processes and learning content were organized and presented (see, e.g., Niederhauser, Reynolds, Salmen, & Skolmoski, 2000). Chou (1999) emphasizes the relationship and similarities between hypertext and instructional theories. He claims that knowledge about how to design instructions is important in order to make use of the possibilities offered by hypertext. Jonassen (1991b) takes it even further and suggests that hypertext and instructional design are a perfect match, and that they are built on similar underlying theories that deal with how to organize information and knowledge. This is likely to be one of the success factors for the use of hypertext within TEL and it is also (together with its relative simplicity) most probably one of the reasons why the web was so quickly adopted for learning and education.

Hypertext provides a clear link between learning and technology and between instructional design and its representation. While instructions were programmatically represented by hard-coding in specialized educational software, hypertext systems, and especially the web, made it possible to represent instructions in a much simpler way that is partly separated from, and independent of, the components that are the building blocks of the content, i.e., instructional units. When html is combined with scripting languages, such as JavaScript, it becomes possible to combine hypertext with simple application logics. However, the representation of pedagogical instruction is still bound to the context of a specific web page, web application or other kinds of applications (such as Flash or Java Applets) delivered on the web. This

²⁸ <http://www.docbook.org/>

problem has later been addressed through specifications such as the IMS Simple Sequencing, and SCORM Sequencing and Navigation to allow a better separation of learning content and pedagogical instruction.

The conditions for delivering TEL and digital learning content changed completely when the web emerged in 1991, even if most people could not anticipate the proportions of the impact of the web at the time. Suddenly it became possible to deliver TEL to anyone at anytime and anywhere in the world – as long as there was an Internet connection available. The markup language of the Internet, html, enhanced the use of hypertext and it became reasonably simple for teachers to design their own learning content and courses on the web, mainly using static html pages at first. User-friendly authoring tools started to emerge in the second half of the nineties, something that boosted the educational use of the web.

In the beginning, the web was static and most of the content was delivered as static html. The level of user interaction was limited to clicking on hyperlinks, a principle that was criticized by Hall (1994) in another hypertext context (see chapter 3.8.1), and this critique is still of relevance. The static nature of the web became a limiting factor and different ways of making the web more dynamic were invented. One of the first approaches was programming using the *Common Gateway Interface (CGI)*. CGI-programs made it possible to execute programs (often coded in C or Perl) at the server and send the results to the client. CGI made simple interactions and interactivity possible by using CGI as a gateway between the web and the application processing the data and vice versa (see, e.g., Boutell, 1996). However, CGI also added a number of bottlenecks and it suffers from performance problems, which led to the development of more sophisticated technologies. To some extent, the performance problems of CGI seem to be theoretical. Studies, such as the one by Varsha, Hansen, & Reeser (2002), shows that the performance of different web programming technologies depends on the type of web application, which means that CGI actually performs very well in some cases. However, one of the more severe problems of CGI is related the http-protocol. The http-protocol is a stateless protocol, which means that it handles every request to the server as a new request and every transaction is independent of previous transactions. This means that the connection to (and state off) the server is lost as soon as a request is taken care of. This becomes a problem, as sessions²⁹ usually need to consist of a number of transactions that are dependent on each other, which makes it harder for programmers to manage complex sessions with multiple interactions over a period of time. At first, this was dealt with directly by programmers using different solutions such as hidden fields and cookies. However, such solutions were not satisfactory because web-based applications were becoming critical, and at the same time were growing rapidly while becoming dependent on other systems, both within and outside a single organization. This increased the need to find solutions that could provide consistency, scalability and stability.

The problems described above have mainly been addressed using two approaches. The first, and most widespread approach has been through the construction of generic application development frameworks, such as Microsoft .Net and Enterprise Java, compared by Sessions (2001), and their corresponding application servers. A second approach (usually combined with the first approach) has been the use of design patterns, such as the *Session Object Pattern*, described by Yoder and Barcalow (1997), for managing sessions states in advanced web applications. Many design patterns have become best practice and are even built into the application frameworks. New constructions to overcome problems caused by the limitations of web-based technology are developed all the time.

²⁹ I have chosen to use the term *session* instead of the term *user session* as sessions containing multiple transactions can be held between two or more systems as well as between a user and a system.

The development described above mainly targets the server-side of web applications, but much development has been done on the client-side as well. The main objective of most of the technologies on the client-side is to find ways to construct rich and responsive user interfaces, which are needed in order to give web applications a similar look-and-feel that users are used to from desktop applications.

While issues on the server, mainly concerning the interaction between server and client, have been addressed with application frameworks, clients are more heterogeneous and harder to handle, because the provider of a service is not in control of the client and problems caused by different web browsers, different software versions, different operating systems and different types of clients, has to be dealt with. Efforts have been made to develop rich interfaces using plug-in based technologies such as Java Applets and Flash, but there is always a problem of being dependent on a third-party plug-in. Plug-ins do not run in all browsers, there are problems caused by different installed versions and so on.

Dynamic web pages and *dynamic html (dhtml)* were introduced in the late nineties as an attempt to tackle some of the limitations of the web. Dhtml made it possible to create dynamic behavior on the client and this made it possible to create slightly richer user interfaces that can be executed in the web browser. Dhtml is a combination of html, the *Document Object Model (DOM)* and JavaScript, which was introduced in 1996 by Netscape, according to Wikipedia (2007). Fairly dynamic web applications could be created using dhtml in combination with server-side technologies such as CGI, Java, PHP and likewise, but combining these technologies with dhtml added to the performance issues discussed above (see, e.g., Iyengar, MacNair, & Nguyen, 1997). JavaScript was later standardized under the name ECMAScript³⁰ (ECMA, 1999), but there were still incompatibility problems and different approaches were needed for different browsers and for different browser versions as their implementation of both html and ECMAScript differed – clearly illustrated by the *Browser Compatibility Chart* from 1999 (Smith, 1999).

Recently there have been consolidations in the development of more structured approaches for rich client functionality and one of the most well-known approaches is *Asynchronous JavaScript and XML (AJAX)*, which can be seen as the successor of dhtml (Paulson, 2005). Ajax is not a single technology, but rather a combination of existing technologies, such as JavaScript, html, XML, CSS and DOM. The basic principle of Ajax is that requests, using *XMLHttpRequest*, to the server and the resulting exchange of data are handled in the background (see, e.g., Garrett, 2005). The effect is that the application feels more responsive and dynamic, as the user does not need to wait for the server to respond upon request. The application logic to manage this is handled using JavaScript and data is transferred as XML. There are however alternative approaches, such as *JavaScript Object Notation (JSON)*³¹ that is an object-based lightweight data interchange format that can be used to serialize data in Ajax applications. The availability of free and open libraries and frameworks for Ajax has increased parallel with the increased interest for Ajax. This, together with emerging best practice and patterns, indicates that Ajax is an interesting option to explore for rich functionality in web browsers. However, it remains to see how the web browser compatibility issue can be handled compared to dhtml.

The possibilities and limitations of web applications, as well as different approaches to deal with those limitations, are of relevance to this thesis as they influence the way in which web-based VLEs are implemented, and hence, also the ways in which the research questions presented in chapter 1.3 can be addressed.

³⁰ ECMA-262

³¹ <http://www.json.org/>

3.9 Metadata and meta-languages

Metadata are essential for addressing all three of my research questions and can be seen as the “glue” that keeps the infrastructure together. The importance of metadata was not primarily a choice that was made from start; it is rather a consequence of the modularization of learning technology. The need for sophisticated metadata boiled down to being an important common denominator for modularity (cf., Erl, 2007; Nagarajan, 2006a; Y W Sim, Wang, Gilbert, & Wills, 2005). This chapter illuminates the underlying theories and concepts for metadata in order to establish a framework for understanding how and why metadata are important for my research and for modularity in general.

3.9.1 *The role and use of metadata*

Metadata are one of the most prioritized areas for learning technology standardization (see chapter 3.6.1). The reasons for this are manifold; metadata are needed in order to describe digital resources (including learning objects), for cataloguing purposes, for use in repositories from where resources can be located and retrieved, for describing specific properties and dependencies, such as technical or pedagogical properties and dependencies, and so on (see, e.g., Haynes, 2004; Duval, 2004). Metadata can be added and used by different actors – both humans and systems. The metadata usage described above is rather frequent and well-known application fields for metadata, and represent a traditional view of metadata as *data about data*. Metadata are regarded as more or less objective descriptions of resources, supplied and maintained by a known and approved metadata provider.

The metadata usages described above are obviously of importance, but other important application fields for metadata are emerging and require more sophisticated approaches. The amount of digital resources are increasing at a rate that makes it important to describe resources and data in ways that are semantically precise to eliminate redundancy as well as for enhancing the abilities for automatization of data processing through techniques such as reasoning. The line between what are metadata and what are data is often ambiguous and there is a need to be able to represent data and metadata in the same way. Even though I use the term metadata when addressing Semantic Web technology it is important to emphasize that Semantic Web technology is applicable to other data as well. This means that machine-readable, and machine-processable semantics are becoming increasingly essential to fulfill those needs, and the Semantic Web addresses such needs (see, e.g., Tim Berners-Lee, Hendler, & Lassila, 2001; Warren, 2006).

Machine-processable semantics are useful for traditional metadata applications as well. Even simple semantics, such as the ability to distinguish between different meanings of common terms, is useful in many situations. The more resources that are available, the more useful machine-processable semantics become. It soon becomes overwhelming to search large amounts of resources without a good management of metadata and an explicit meaning of common terms. There is a risk that users drown in the amount of information that becomes the result of badly designed queries in combination with poor metadata.

As an example: when searching the Library of Congress book catalogue using the word “plane” a result-set containing 5709 catalogue entries is returned. Those entries contain resources spanning over a large number of subjects, all with their own meaning of the word “plane”, such as plane as: plane surface, airplane, level, sailing, plane down, smooth, etc. If metadata (and query) semantics would be expressed in a machine-processable way, the query could be limited to only comprise one meaning of the word “plane” and by that limiting the number of returned records. This is just one simple example of how machine-processable metadata semantics are useful. Semantics are related to the context in which it is used and to

the ontology that is associated with that specific context. A lightweight ontology approach to overcome some of the drawbacks of traditional directory-based approaches, by adding formal semantics, is suggested by Zaihrayen et al. (2007).

In a learning situation, metadata often need to be provided and maintained by different users with different roles. An example of this is given in paper 6, where teachers and students are encouraged to contribute to the metadata that are related to curricula. This transforms metadata from the traditional view as being authoritarian and regarded as objective, to also being democratic, context dependent, and subjective, evolving in a living and constantly changing eco-system. Nilsson, Palmér, & Naeve, (2002) refers to this as a *metadata ecology*. The technology that underlies the Semantic Web appears as suitable for an ecological approach to metadata (see, e.g., M Nilsson, Palmér, & Naeve, 2002; M. Nilsson, 2001; M Nilsson, Naeve, Duval, Johnston, & Massart, 2008; Tim Berners-Lee, Hendler, & Lassila, 2001; Palmér, Naeve, & Paulsson, 2004; Shadbolt, Hall, & Berners-Lee, 2006).

Shadbolt, Hall, & Berners-Lee (2006) emphasize that machine-processable semantics becomes increasingly essential when there is a need to connect different pieces of data and metadata based on their semantics. Machine-processable semantics facilitate information integration from different and independent sources using a common standard for representing data, i.e., the *Resource Description Framework (RDF)*. Information integration can be accomplished using different techniques for matching and filtering, ranging from simple matching of terms, to the use of advanced logics and reasoning based on formal ontologies (see, e.g., Fensel, Van Harmelen & Davies, 2003).

Beside the use of RDF for data integration, the ability to describe components and services using machine-processable semantics is essential in modular environments (see, e.g., Nagarajan, 2006a; Nagarajan, 2006b; Lama, Arroyo, Sicillia, & López-Cobo, 2006; Yu, 2007). Components, in terms of services and functionality, need to be described in different ways depending on the context in which they are used. Different descriptions are needed for different purposes and in different contexts. By adding machine-processable semantics to service descriptions it becomes possible to dynamically match services (i.e., components) with other services for coupling into composite applications based on the semantically expressed qualities and characteristics (see, e.g., Nagarajan, 2006b). This is the objective of *Semantic Web Services* (see chapter 3.11.4).

3.9.2 Representing machine-processable semantics: the Semantic Web

Semantic Web technology is the most common way to represent machine-processable semantics on the web. The backbone of the Semantic Web is RDF, a specification that is maintained by the *World Wide Web Consortium (W3C)* as a part of the *Semantic Web Initiative*³². In simple terms, RDF is a formal specification of a standardized way of representing data and data models. By using a standard³³ and common design principles other RDF-aware applications can interpret and process the data model without prior awareness of that specific model (see, e.g., Ekengren, 1998; Antoniou & Van Harmelen, 2004).

The idea of a Semantic Web was first expressed by Tim Berners-Lee & Fischetti (1999) and originates from the idea of adding a machine-processable semantic layer on top of the current human-processable web (see, e.g., Antoniou & Van Harmelen, 2004, pp. 16-17). The reason for such an approach is that the web makes sense only to humans but not to machines. Hence, the information on the web can be read and interpreted by humans but computers cannot make any sense of the information because there is no way of separating the data and data

³² <http://www.w3.org/2001/sw/>

³³ Technically RDF is a W3C recommendation

semantics from its presentation and structure. Even in cases when it is possible to separate the data, for example using XML, the semantics of the data is still only expressed in human language and is highly context dependent. Hence, the Semantic Web adds a formal model for expressing semantics that can be interpreted and used by computers to extract meaning from the web. This allows information on the web to be automatically collected and processed by computers. Semantic Web technology makes it possible to integrate data from different, otherwise independent, data sources without converting between data formats (see, e.g., Warren, 2006).

The use of structured data formats on the web has increased since the introduction of XML in 1998 (see, e.g., Bray, Paoli, & Sperberg-McQueen, 1998). XML provides mechanisms for representing machine-readable data on the web in a way that separates data and presentation of data using style sheets. This means that the mechanisms for separating data, syntax, and presentation are already available. However, it should be re-emphasized that XML was designed to represent syntax and not to represent semantics (see, e.g., Tim Berners-Lee, 1998; Decker et al., 2000). In that respect, RDF and XML complement each other, and it is common practice to represent RDF using XML (RDF/XML) (see, e.g., Beckett, 2004) in order to take advantage the machine-readable capabilities of XML and related technologies.

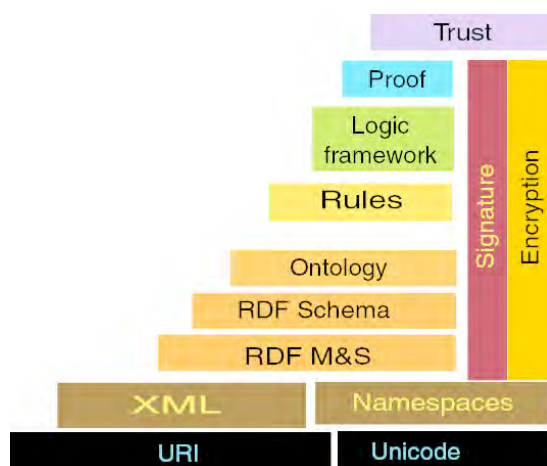


Figure 3-5. The layers of the Semantic Web and their mutual relationship (Hazael-Mas, 2003)

By adding a *semantic layer*, which actually consists of several semantic layers (see Figure 3-5), to the existing web, it becomes possible for computers to read, interpret and process information on the web. This creates possibilities for new kinds of services that do not only rely on traditional search engines (such as Google³⁴), but that use logics and reasoning in order to analyze and optimize the use of information, combining and integrating information and data from multiple sources, as well as utilizing personal preferences for adaptation purposes. The foremost added value of the Semantic Web is the possibility to deliver new and advanced services on the web based on information and service integration combining multiple sources on the fly (see, e.g., Tim Berners-Lee, Hendler, & Lassila, 2001; Shadbolt, Hall, & Berners-Lee, 2006; Warren, 2006; Nagarajan, 2006b).

³⁴ <http://www.google.com/>

3.9.3 The Resource Description Framework

RDF is the core technology of the Semantic Web, and the only serious competitor to RDF is *Topic Maps (TM)* (Pepper, 2000). Even though TM and RDF are quite comparable, RDF is the technology that is most widely adopted on the web. However, there are ongoing initiatives to harmonize the use of RDF and TM (see, e.g., Garshol, 2003; Pepper, Vitali, Garshol, Gessa, & Presutti, 2005)). The choice of using RDF was made based on its wide adoption and on the availability of suitable tools and code libraries for RDF.

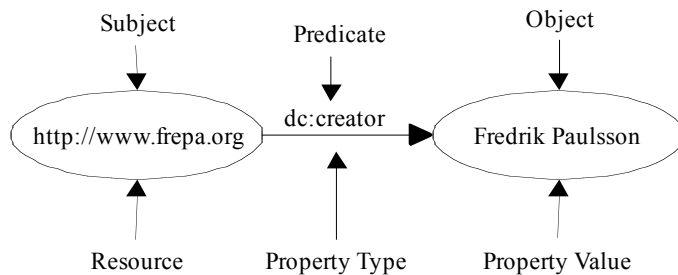


Figure 3-6. A simple RDF-graph expressing that: Fredrik Paulsson is the creator of the resource <http://www.frepa.org>.

RDF has a fairly simple model for expressing machine-processable semantics. The RDF-model is based on *object-property-value* triplets and a triplet can be represented by a directed graph, illustrated in Figure 3-6. The graph view of the RDF representation is usable for illustrating the basic principles of RDF.

Figure 3-6 illustrates how a statement is made about a *resource*, in this case my website at <http://www.frepa.org/>. The website (*the subject*) is uniquely identified using a *Universal Resource Identifier (URI)*. All properties and resources must be uniquely identified using an URI. The subject has a *property* “created by” (using Dublin Core) that is valued by the *object*. In this case the value is the creator: “Fredrik Paulsson”. All together this is called a *RDF statement*, (Hayes, 2002).

The property value in the example could also be represented using a domain specific machine-processable *ontology* for representing humans and human relationships in social networks, such as the *Friend Of a Friend (FOAF)*³⁵ (see, e.g., Brickley & Miller, 2007), which its use is discussed by Li, Lina, Finin, and Joshi (2005). FOAF is defined using *RDF Schema (RDFS)* (Brickley & Brickley, 2004), and the Web Ontology Language (OWL) (McGuinness & van Harmelen, 2004). While RDF is used to represent the data models there is also a need for ways of representing predefined domain models as ontologies. FOAF is an example of an ontology for the domain of social networking. The concept of domain ontologies, which originates from the *Artificial Intelligence (AI)* community, was later described by Gruber (1993) as a *social contract*. On the Semantic Web, ontology can be defined as *a machine-processable collection of concepts and their relationships expressed using a common language in order to create an abstract model of a specific domain*. Defining classes, properties and their mutual relationships is needed to accomplish this. RDF Schema has limited expressive power as an ontology description language and OWL was developed to overcome this by providing a powerful ontology language for the Semantic Web (Horrocks, Patel-Schneider, & van Harmelen, 2003). Ontologies are often discussed in terms of formal ontologies, but ontologies can also be of a less formal character, as illustrated in Figure 3-7.

³⁵ <http://www.foaf-project.org/>

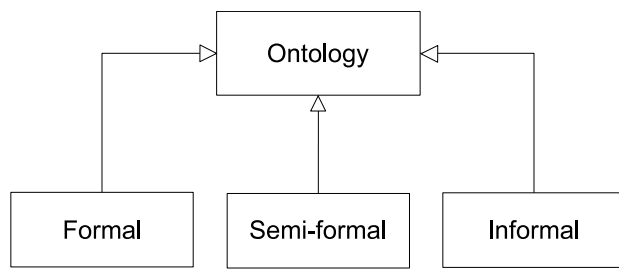


Figure 3-7. The relationship between different ontologies.

The expressive power of the statement in Figure 3-6 becomes much more powerful if the creator is represented using the FOAF ontology instead of a string. FOAF contains much more information that can be accessed and extracted; as a part of the RDF-statement in Figure 3-6 or as a part of the Semantic Web metadata ecology for use in new contexts. Grimnes, Edwards, & Preece, (2004) point out how such ontology-based RDF statements, using the FOAF ontology, can be used for creating adaptive and dynamic social networks.

Even though the RDF-model is simple, it is capable of expressing sophisticated semantics that can be used to extract a fair amount of machine-processable meaning. This is exemplified by the use of FOAF in social networks (see, e.g., Grimnes, Edwards, & Preece, 2004; Li et al., 2005). It is possible to express complex semantics about any resource by combining several simple RDF statements (see Figure 3-8).

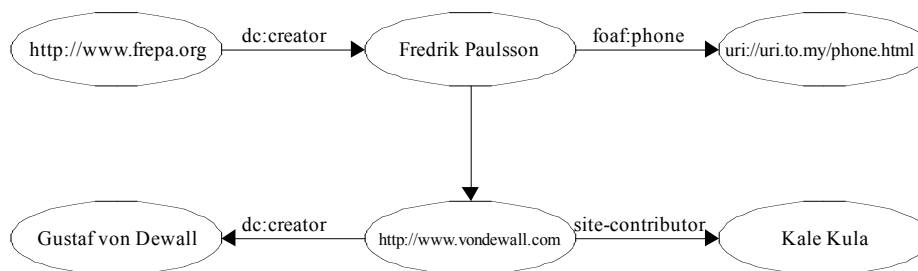


Figure 3-8. Several RDF statements can be combined to express sophisticated machine-processable semantics.

The Semantic Web is a fairly new phenomenon that rests on a technology base which, in many respects, is still uncharted territory. There are still many issues to be addressed and problems to be solved.

3.9.4 RDF and XML

As emphasized in previous chapters, the purpose of RDF is to represent semantics while the purpose of XML is to represent syntax. This is an important distinction to make in order to understand the relationship between RDF and XML (see Figure 3-5). A common misbelief is that RDF is a competitor to XML and that they are contradictory because of that. This is however not true - RDF and XML complement each other, and one of the strengths of RDF is its ability to be represented using XML. As such, XML is the carrier of syntax, i.e., says nothing about information semantics only about the structure, while RDF is the carrier of

information semantics. Listing 3-1 gives an example of how RDF is represented using RDF/XML. The XML representation of RDF is called RDF/XML, and it is the standard way to represent RDF data in RDF aware systems (see, e.g., Beckett, 2004). One important reason for the use of RDF/XML is that it makes it possible to utilize several powerful XML technologies together with RDF. Such an example was given when addressing research question 3 (see papers 5 and 6), where XPointers and XML DOM were used to determine identifiers for RDF annotations.

```
<?xml version=1.0" encoding="UTF-16"?>
<rdf:RDF
  xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
  xmlns:frepadomain="http://www.frepa.org/rdf-ns#">
  <rdf:Description rdf:about="http://www.frepa.org">
    <frepadomain:site-creator>
      Fredrik Paulsson
    </frepadomain:site-creator>
  </rdf:Description>
</rdf:RDF>
```

Listing 3-1. An XML-representation of the RDF-graph in Figure 3-6 (without the ontology references).

3.10 Meta-languages for learning design

When TEL evolved, and the focus changed from specialized training software and CD-ROM to the Web, a need arose to separate the representation of pedagogical design and process from the VLE and the learning content.

A common approach to accomplish this separation has been through the use of special purpose meta-languages, also referred to as *Educational Modelling Languages (EML)*³⁶ by Rawlings, Koper, van Rosmalen, Rodríguez-Artacho, and Lefrere (2002). Generic meta-languages for representing pedagogical instruction and process are a rather new phenomenon, and such representation has traditionally been hard-coded into digital learning content and educational software, which can be argued to be a type of digital learning content. This is, for example, the case in most CD-ROM productions.

EMLs have emerged during the last 10 years and there are several different EMLs available. The purpose of an EML is to provide the means for machine-readable representations of pedagogical design and pedagogical process. In this thesis, the term EML is used as a general and collective term for all special purpose meta-languages for TEL, as defined by Rawlings et al., 2002) (see below).

Some EMLs, such as the IMS Learning Design (IMS, 2003a), are developed from a general-purpose perspective, while others originate from a certain discipline and are developed for a specific purpose, such as the *eLesson Markup Language (eLML)* (Fisler & Bleisch, 2006). The eLML was developed by the *Geographic Information Technology Training Alliance*

³⁶ Not to be confused with the Educational Modelling Language that was developed by the Open University of Netherlands (OUNL-EML), and refers to their specific implementation of an educational meta-language.

project (GITTA) and addresses learning in, and about, *Geographic Information Systems (GIS)*. There are a number of proprietary EMLs as well, but in this thesis I have chosen to regard EMLs as something that must be open, non-proprietary, and preferably standardized by an accredited organization. This is motivated by that much of the idea of an EML is lost if it is made proprietary.

An extensive survey of six EMLs was provided by Rawlings, et al. (2002). The survey gives a good idea about the concept of EMLs that is defined as:

“...a semantic information model and binding, describing the content and process within a ‘unit of learning’ from a pedagogical perspective in order to support reuse and interoperability.”

The definition is general enough to include most of the properties that are expected from an EML.

It can be argued that specifications such as the SCORM Navigation and Sequencing (Capone, et al., 2004) and IMS Simple Sequencing (IMS, 2003b) do not qualify as EMLs, even though they implement similar functionality and are partly used for the same purposes. The main argument is that they do not include the pedagogical semantics that are needed for representing pedagogical instruction, and they are mainly being used for simple aggregation and sequencing (see, e.g., Rawlings et al., 2002; Koper & Manderveld, 2004).

In their study, Rawlings et al. (2002) point out that even though there are similarities between the six surveyed EMLs, there are also differences in how they comply with the definition above. Their conclusion was that there was only one EML that fully complied with the definition and that was the OUNL-EML. For this reason the OUNL-EML was used as the basis for the development of the IMS LD that is discussed below.

3.10.1 IMS Learning Design

Even though the EML survey by Rawlings et al. (2002) provides a good overview of EMLs, the fast pace of development during the last couple of years has made it a bit outdated. EMLs are consolidated and one reason for this is that the OUNL-EML, which was found to be the most versatile (as well as extensive) in the survey, was adopted by IMS as the basis for the development of the IMS LD (Koper, 2005). This is likely to give IMS LD a unique position as the EML most likely to be widely adopted by LMS vendors, at least in the foreseeable future.

The objective of IMS LD is to be general enough to describe and represent any pedagogical approach. Koper and Manderveld (2004) refer to this level of abstraction as a *pedagogical meta-model*, i.e., an abstract model for representing pedagogical design, which they refer to as *Learning Design*. IMS LD provides a data-model, a conceptual model, and the meta-language needed for expressing pedagogical models and their associated processes in a machine-readable format. This is done using a formal notation together with an XML-binding. However, the XML-binding results in the same kind of problems as with XML-based metadata that were discussed in chapter 3.9. The problems are similar and related to the different natures of graph-based versus document-based metadata, which are unable to represent machine-processable semantics in a standard way. Hence, there is a risk of preserving some of the problems associated with representing pedagogical design and pedagogical process, discussed in chapter 3.4.

An important characteristic of IMS LD is that it does not have an explicit single learner focus or an explicit group learning focus. This means that IMS LD supports group activities and collaborative activities, as well as single learner activities. One effect of this, together with the

ambition to support a multitude of pedagogical approaches, is that it suggests a shift of focus, from pre-packaged, self-phased learning, to learning that takes place in a social context, and learning activities that involve groups of learners as well as individual activities and processes (e.g., Koper & Tattersall, 2005). This shift in focus is also intimated by a shift in terms. Instead of using the term “instructional design” the term “learning design” is consequently used when refereeing to pedagogical design in relation to IMS LD. Even though IMS LD and other EMLs originates from instructional design theory, IMS LD indicates a shift toward approaches that focus more on learning and less on training and performance.

The metaphor of a play or a film script can be used to describe how IMS LD works (see, e.g., Koper, 2005; Olivier & Tattersall, 2005). As such a metaphor implies; a learning design is constructed around a number of roles held by different actors in the system. Based on their respective roles, actors go through a number of activities within the setting of the script. These activities are either sequenced or based on the simple logic that is allowed by IMS LD.

The script metaphor works well for illustrating the main characteristics of IMS LD, but it also illustrates one of the main weaknesses, which can be argued to be a legacy from instructional design and the representation of instructional design in VLEs. The fact that the pedagogical instruction has to be predetermined and explicitly expressed in advance, which is also the case with a film script, does not leave much room for improvisation or too extensive discrepancies. There is a clear purpose and an intrinsic value in explicitly outlining the whole chain of events in advance when working with films or plays, but for learning this is limiting in a way that may restrain the pedagogical flexibility. However, in the case of IMS LD, it should be emphasized that the XML-binding is an important contributor to this problem, and the fact that IMS LD is based on an abstract model creates the means to deal with this issue. Koper (2004) suggests an alternative approach, based on Semantic Web technology, in order to solve some of the issues (see chapter 5 for a detailed discussion).

The structures and interrelations of the activities and processes encountered in learning situations are often complex and thereby hard to represent in a flexible enough way, using existing EMLs, and the representation of learning design, pedagogical processes and structure are still expected to be predetermined. This is something that is often hard to deal with – considering the nature of constructivist learning theories (see chapter 3.2).

3.11 Component technology and modularity

This chapter reviews some of the developments and theories within *Component-Oriented Software Technology*. I will also make comparisons and draw parallels to the modular developments within TEL, and especially related to learning objects.

Computer science has strived toward reuse and interoperability through modularity for many years. The most well-known and used approach is likely to be Object-Oriented programming, which has become the de facto standard in the software industry. While OOP facilitates reuse and interoperability at a micro level (that is, within software projects), at a programming language level and at an organizational level, there is also a need for reusability and interoperability at a macro level, i.e., across organizations, across programming languages and across different software and hardware platforms.

The development of OOP and the endeavor toward modular approaches eventually led to the evolvment *Component-Oriented Software Engineering*. The success of Component-Oriented Software Engineering depends, like OOP, on the decomposition of software systems. Successful decomposition of software systems results in independent but interacting software components that together make up the system by communicating and interacting through

well-defined message interfaces. Szyperski and Pfister (1997) define a *software component* as:

“A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties.”

Nierstrasz and Dami (1995) defines a component in a similar way, but much shorter, as a “...*static abstraction with plugs*”. While Szyperski and Pfister (1997) address both technical properties and business properties in their definition, Nierstrasz and Dami only address technical properties. A component as a unit is characterized by that it is deployed independently, it can be used for composition by a third-party, and it has no externally observable state (Szyperski, 2002c). Those characteristics influence the ways in which components can be developed and it makes components differ from object in important ways. Szyperski (2002c) points out that objects need a predetermined “construction plan” (i.e., a class) to be initiated, describing the initial state and behavior of the created object, while components do not. Szyperski (2002a) suggests a categorization of components into *blackbox*, *greybox*, respectively *whitebox* components. Szyperski bases his categorization on the visibility of the implementation details of components. Blackbox components completely hide implementation details, exposing only interfaces, while whitebox components’ implementation details are fully visible, and greybox components are somewhere in between. Nierstrasz and Dami (1995) distinguish between objects and components by emphasizing that a component can encapsulate any useful software abstraction, while an object only encapsulates data and behaviors. In fact, components’ internals do not need to be implemented using an OO-approach. Nierstrasz and Dami (1995) distinguish between *methodological aspects* of components and *technical aspects* of components. This distinction is interesting for two reasons: It separates conceptual aspects of components from technical aspects and technical quality issues, and it provides an interesting point of view for comparing characteristics of software components with the characteristics of learning objects – a discussion that is continued in chapter 5.

Software components are intended to be used as building blocks for the composition of *composite applications*, and as such they are not necessarily useful in isolation. Software components are contributing to an added value when they are used as parts in the wider context of a larger “machinery”, i.e., a composite application (see, e.g., Nierstrasz & Dami; 1995; Szyperski, 2002b; Weinreich & Sametinger, 2001).

While traditional software is mainly static, a modular (composite) application can change at run-time. Components in a composite application can be regarded as self-standing, independent parts that communicate through messages. This becomes even more pronounced in *Service-Oriented Computing* (see chapter 3.11.2). Nierstrasz and Dami (1995) make a distinction between *composition-time* and *run-time*, but emphasize that composite applications may be recomposed at run-time based on input from other parts of the system or on external input. These are characteristics that are relevant for the design and implementation of modular VLEs, which was demonstrated in papers 3 and 4, and for the interrelation of the components of a modular VLE (also demonstrated and discussed in papers 3 and 4), as well as for the representation of pedagogical design and pedagogical process, which was demonstrated in papers 5 and 6 by relating and connecting learning objects and the curriculum through semantic annotation (see discussion in chapter 5).

Even though the idea of software components may seem simple enough, the reality of composite software is complicated. Components need to work together independently of vendor and preferably also independently of software or hardware environment, but this has

turned out to be far from simple to accomplish. A basic condition for this to happen is the existence of standards (see, e.g., Szyperski, 2002b). Standards and interoperability are often associated with trade-offs and added complexity. Much of the adoption of component frameworks has occurred in non-vendor neutral frameworks, such as the frameworks from Microsoft³⁷ and SUN³⁸, and not in frameworks based on formal standards, such as the work by the *Object Management Group (OMG)*³⁹.

As pointed out in chapter 3.6, the original source of inspiration for learning objects was Object-Oriented software development. However, when regarding the properties ascribed to learning objects compared to the properties and definitions of software objects, respectively software components, it boils down to the conclusion that learning objects actually have more in common with software components than with software objects (cf. Nierstrasz & Dami, 1995). Learning objects would probably be better off compared to components than objects: i.e., *learning components* seems to be a better term to describe the intended characteristics of learning objects.

3.11.1 Component frameworks

There are several existing component frameworks, such as Microsoft's platform, with technologies like *COM*, *COM+*, *Common Language Infrastructure (CLI)* and *ActiveX*, the *Java platform*, with technologies like as *JavaBeans*, *Enterprise Java*, and *Remote Method Invocation (RMI)* and OMGs *Object Management Architecture (OMA)* and the *Common Object Request Broker Architecture (CORBA)* that are platform neutral. The Microsoft framework and the Java framework seem to be the two most used and referred to frameworks, while OMA and CORBA appear to be having a hard time. Even though CORBA can be used for bridging between the other two platforms, it is often criticized for being too complex, which makes it expensive and complicated to implement (see, e.g., Maffeis & Schmidt, 1997; Henning, 2006). These are problems that are likely to occur when the trade-offs are not compensated by other advantages, which is the case with Web-Services where trade-offs in performance and efficiency are counterbalanced by advantages, such as global interoperability and business advantages (e.g., Henning, 2006).

The Java framework is platform neutral as well, but Java is not vendor neutral and it is not a standard in the same sense that CORBA is. Since version 3.0, CORBA is adopting technologies from EJB and there is a harmonization going on between the two (Szyperski, 2002b).

While these component frameworks address software components on the client, as well as software components at the server, there is another modular approach that has boomed during the last decade, and that is *Service-Oriented Computing*, and especially implemented as *Service-Oriented Architectures (SOA)* and *Web-Services*.

3.11.2 Service-Oriented Computing

The breakthrough of the Internet and the web has changed the nature of computing (see chapter 3.8.3). This is an ongoing change where modularity and distribution of the infrastructure play a central role. The perception of computer systems and their demarcations changed as the Internet started a globalization of systems that has eventually led to an evolutionary change in how systems are designed and implemented. SOA and Web Services play an important role in this change. One of the most notable changes in system design

³⁷ <http://msdn2.microsoft.com/en-us/library/aa139693.aspx>

³⁸ <http://java.sun.com/>

³⁹ <http://www.omg.org/>

practice is that software logic is decomposed and packaged into manageable units for distribution as services. This change affects new system development as well as old legacy systems that are being modernized and equipped with service interfaces that at least create an illusion of modularity to the surrounding world. The effect expresses itself in several ways. One of the tendencies that can be seen is that it forces the design away from SOA in terms of *Silo-Oriented Architectures*, in favor of SOA in terms of *Service-Oriented Architectures* that provide the means for horizontal communication and interaction that take place *between* systems and loosely coupled services (see, e.g., Patrick, 2005; Arsanjani & Holley, 2006).

Erl (2007, pp. 29-30) argues that Service-Oriented Computing is a *design paradigm*, in the same sense as Object-Oriented is a design paradigm, with its own design characteristics, architectural models, as well as design standards and conventions. Erl also emphasizes that Service-Oriented Computing is a comprehensive topic, and one of the reasons for this is that it builds on previous work on distributed computing, adding new layers and ideas that are better adapted to present needs. He also makes a distinction between Service-Oriented Computing and SOA: Service-Oriented Computing consists of a number of elements that are the parts that constitute a *Service-Oriented Computing platform*, and SOA is only one of those elements. Other elements are *Services*, *Service Composition*, *Service Inventory*, etc. Service-Oriented Computing is the paradigm and SOA is the *technology architecture model* that implements Service-Oriented Computing. Figure 3-9 shows a conceptual view of how the different elements of Service-Oriented Computing interrelate.

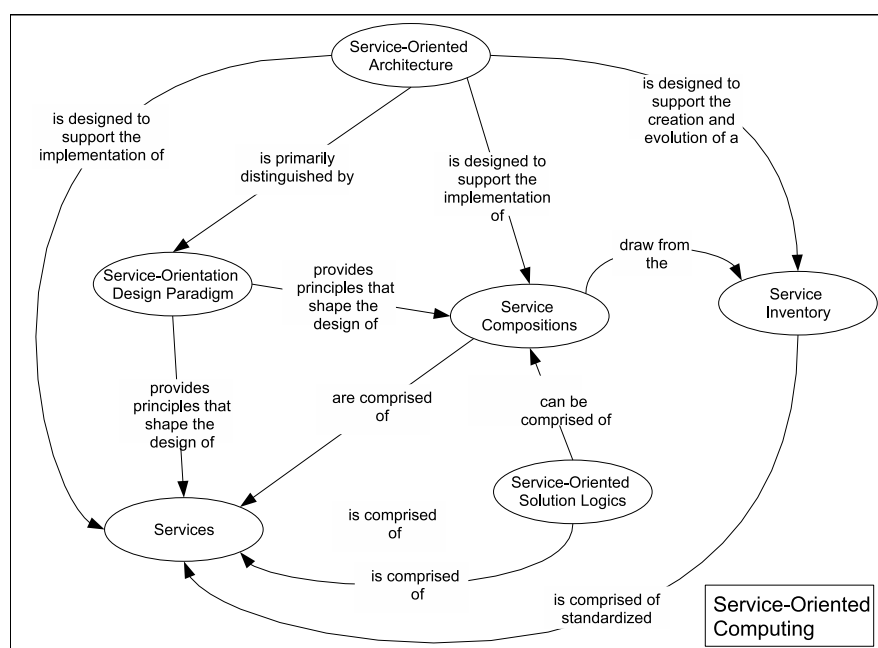


Figure 3-9. A conceptual view of how the different elements of Service-Oriented Computing interrelate (based on the figure by Erl (2007, p. 41).

Even though Earl's description of Service-Oriented computing is clear and semantically relatively unambiguous it has become common to use the term SOA in a wider sense. Sim, Wang, Gilbert, and Wills (2005) define SOA as a "...loosely couple network of communicating services", while Leymann, in the foreword of Cardoso & Sheth (2006a), defines SOA as "fundamentally it describes how service requesters and service providers can be decoupled via discovery mechanisms resulting in loosely coupled systems". Even though it

may contribute to the confusion and inconsistency that is sometimes associated with the term SOA, it is quite a common practice. SOA is, concurrently with its growing popularity, sometimes even used synonymously with the term “Web Services”. It is however important to emphasize that, in this thesis, Web Services are regarded as a *technology platform* that can be used for the implementation of SOA, which in turn is the architecture model upon which Web Services are implemented (Erl, 2007). SOA can be implemented using other technology platforms as well. For example using one of the widespread component frameworks associated with Enterprise Java, as demonstrated in paper 3, or Microsoft .Net (without adhering to the Web Service technology platform) or using alternative architectural styles, such as *Representational State Transfer (REST)* architectures discussed by Fielding (2000, pp. 76-106). However, proprietary solutions appear to be decreasing concurrently with the penetration of Web Services and the increasing requirements for interoperability caused by the need of global interaction and distribution of computing that was ultimately brought on by the adoption of Internet and web-based technologies.

There are several similarities between Service-Oriented and Component-Oriented. One of the most notable similarities is the emphasis on software composition, rather than on programming. Services are the subject of aggregation through a process of composition in a similar manner as with software components. A service is provided as an encapsulated and *loosely coupled* software component. Loose coupling is one of the most important characteristics of SOA, because it is the property that separates data from the processing of data, and with that the type of service from the software providing the service (see, e.g., He, 2003). This includes hiding the implementation details of a service from the consumer of a service. Loose coupling is also important for modularity, as services can be “pooled” together through composition in order to form new functionality in new contexts – such as in the context of a VLE, involving various levels of complexity. Hence, aggregating and combining multiple services accomplishes new services, providing functionality with an added value, and it is the composability of services that makes them reusable. Loose coupling strives to reduce dependencies between the service, how it is implemented, and the consumers of the service. The level of coupling is related to the abstraction level of a service, which in turn is affected by how efficiently the internals and details of a service are hidden when the service is provided to consumers. This also influences the level of reusability and composability of a service – the characteristics that create flexibility (see, e.g., He, 2003).

3.11.3 Web Services

Web Services is the technology platform that has become the most widespread for implementing SOA on the Internet, and has quickly gained the status of being an industry standard. The Web Service technology platform seems to have evolved as an extension of the developments on the web that were discussed in chapter 3.8.3, and can be argued to represent a more structured and standardized approach to web application development. The initial hype has evolved into real-life production solutions concurrently with that the technology platform has matured and stabilized.

Web Services are implemented and exposed as services by a Service Provider. A Web Service can implement newly developed functionality or functionality that resides in legacy systems and that is provided with a Web Service interface to allow the exposure of functionality to other systems. Services are located via a Web Service directory, usually based on the *Universal Description Discovery and Integration (UDDI)*, in order to be used by a *Service Consumer*. The discovery is dependent on metadata descriptions of the available services. The Web Service cycle is illustrated by Figure 3-10.

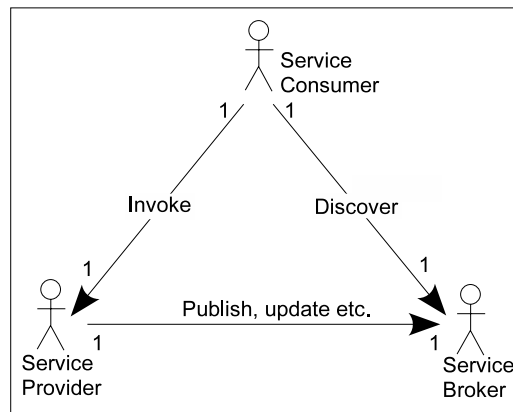


Figure 3-10. The Web Service cycle.

The technical foundation for implementing Web Services is a stack of standards that are needed throughout the service lifecycle (Szyperski, 2002b). These standards and their respective use are described in Table 3-4.

Usage	Specification/protocol	Comment
Discovery	UDDI	Itself a Web Service, UDDI serves as a directory for Web Services.
Description	WSDL, WSFL/XLANG, others to come	Given one or more Web Services, describe properties at a meta-level.
Access	SOAP, SOAP with attachments	Given a Web Service instance, access it via messages.
Transfer	HTTP, SMTP, others	Transfer SOAP-messages, incl. attachments.
Transport	TCP/IP, UDP, and others	Protocols for data transport.

Table 3-4. The core Web Service protocol stack, based on the table by Szyperski (2002b, p. 229).

3.11.4 Semantic Web Services

The importance of metadata is emphasized in most literature on Service-Oriented Computing, (see, e.g., Cardoso & Sheth, 2006a; Szyperski, 2002a; Erl, 2007; Yu, 2007). Services need to be thoroughly described for discovery purposes, as well as for composition purposes. The functionality of services and service interfaces needs to be described in enough detail to avoid any ambiguousness at discovery-time, composition-time, or at run-time. This means that standards such as the *Web Service Description Language (WSDL)*, in combination with UDDI are crucial. However, the more heterogeneous the environments become, the more sophisticated metadata are needed and it is not enough to have access to syntactic heterogeneous metadata, there is also a need for *semantic heterogeneity* (Cardoso & Sheth, 2006a).

In order to address issues of semantic metadata heterogeneity for Web Services there is ongoing work to develop what is referred to as *Semantic Web Services*. In simple terms, Semantic Web Services are a combination of Web Service technology and Semantic Web

technology in order to create Web Services with semantic capabilities for use on the Semantic Web. While Web Services act as components in self-describing and self-contained applications that are described using WSDL, WSDL provides only syntactic operational information of the Web Service and no semantic information (see, e.g., Cardoso & Sheth, 2006a); Yu, 2007, pp. 207-217). By combining Web Service technology and Semantic Web Technology, for example, through the application of semantic annotations to Web Services, and connecting them to ontologies that provide explicit semantic definitions, the descriptions of Web Services become significantly more powerful (see, e.g., Nagarajan, 2006a). There are several initiatives suggesting specifications for use with Semantic Web Services. Examples are the OWL-based OWL-S⁴⁰, the *Web Service Modeling Ontology (WSMO)*⁴¹, the *Semantic Web Services Framework (SWSF)* (Battle et al., 2005), and the WSDL based *WSDL-S*⁴² (Cardoso & Sheth, 2006b). Semantic annotations of Web Services affect both the descriptions of Web Services, the use of Web Services, and the discovery of Web Services. The latter meaning that machine-processable semantics facilitate the discovery of Web Services by helping the matching of processes and requirements to Web Services, and by that facilitating the automatization of Web Service composition (see, e.g., Klein & Bernstein, 2001; Rodriguez & Egenhofer, 2003). An effect of these added characteristics is that the adaptable nature of composite applications based on Web Services extends to the potential of being *adaptive* and not only *adaptable*. This is a characteristic that I argue to be essential in order to achieve modular VLEs that are pedagogically neutral and that can respond to semantic representations of pedagogical design (see discussion in chapter 5).

3.12 Emerging frame of reference

The objective of this chapter was to elucidate major theories, concepts and technologies that influence TEL and the implementation of learning technology. The purpose has not been to take a stand or value learning theories or other pedagogical theory. The objective was to show that there are several available learning theories and emphasize their similarities and differences in order to establish a relationship to the implementation of learning technology and the different requirements that need to be met in order for learning technology to be reasonably neutral to pedagogical theories. In order to understand those relationships there is a need to be aware of the variety of components that influence TEL.

The focus of this thesis is on system and software architecture and the role of modularity in VLEs and digital learning content. However, due to the nature of the research presented in this thesis, the theoretical frame of reference will be used to illustrate and illuminate the interdependence between pedagogical theory, in terms of how it affects pedagogical design and pedagogical process, and how it also affects the system representation of pedagogical process, which leads to effects for the design and implementation of learning technology and digital learning content. This will be discussed in detail in chapter 5.

Figure 3-11 summarizes the theoretical frame of reference that has been described and discussed in this chapter.

⁴⁰ <http://www.w3.org/Submission/OWL-S/>

⁴¹ <http://www.wsmo.org/>

⁴² <http://www.w3.org/Submission/WSDL-S/>

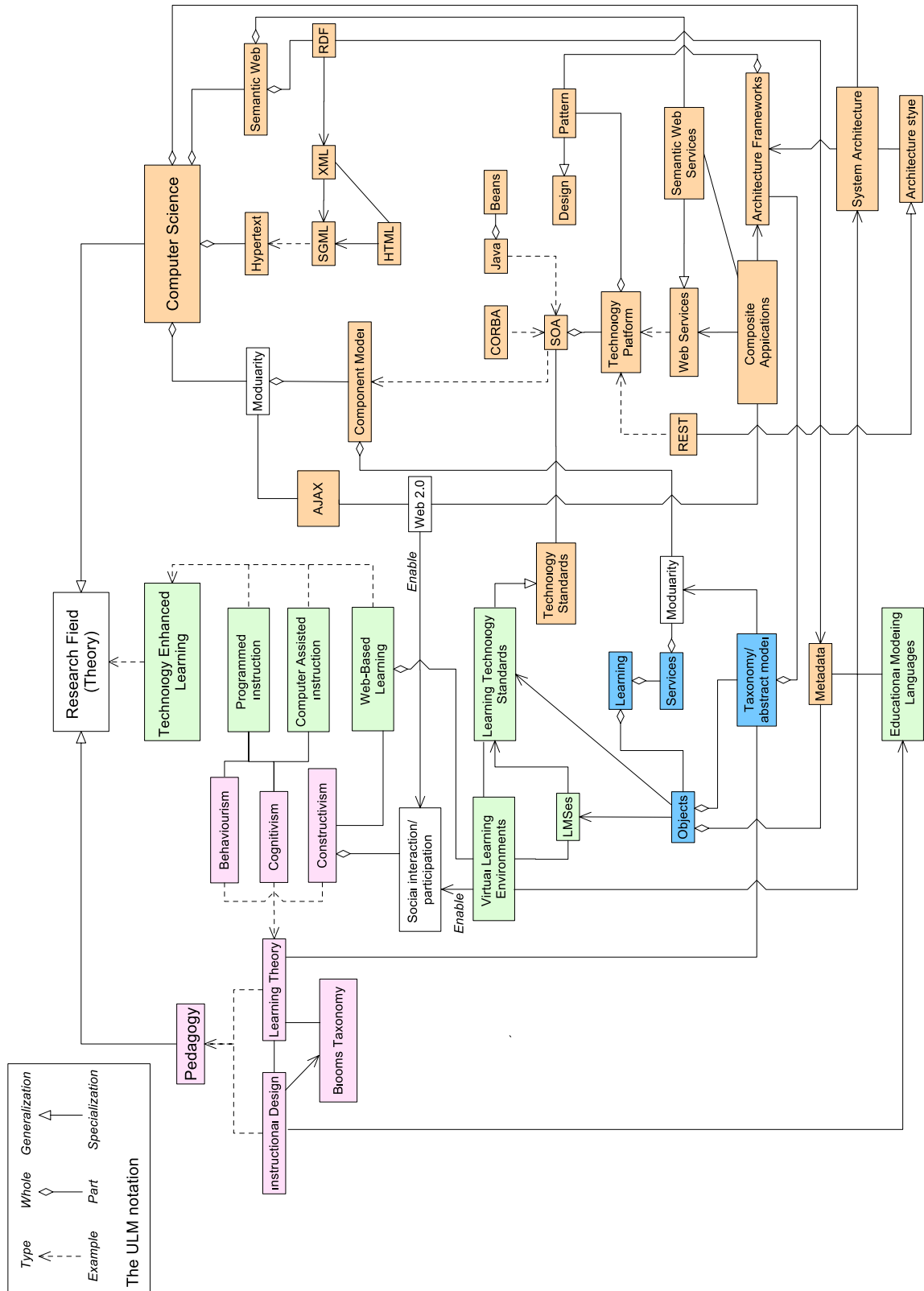


Figure 3-12. Summary of the theoretical framework for the research presented in this thesis.

4 Summary of research results

This chapter summarizes the research results presented in the six appended papers. The results are further discussed and analyzed in chapter 5.

The research questions formulated in chapter 1 were:

1. *How can a common abstract model for modularity, which covers both the VLE and digital learning content, be accomplished without creating incompatibilities with existing models?*
2. *How can a technical infrastructure be designed and implemented according to a common abstract model, in order to provide the necessary supporting infrastructure, and without disqualifying legacy?*
3. *How can pedagogical designs and processes be represented in order to utilize modularity to increase pedagogical flexibility and adaptability?*

These research questions were addressed in the appended papers, and the research results presented in papers 1-6 are summarized below and discussed in chapter 5.

4.1 Results of paper 1

“Establishing technical quality criteria for Learning Objects”

Paper 1 examines definitions, designs and implementations of learning objects. The study consisted of two parts: a comparative study of learning object implementations, and a meta-study comparing learning object definitions. The results were analyzed from the point of view of *technical quality criteria*. Parts of the contents (in total over 200 learning objects) from three different learning object repositories (LOR) were studied⁴³, focusing on technical implementation and software architecture issues. The studied learning objects were estimated to be representative for how learning objects are commonly designed, implemented and distributed – even though there are of course exceptions.

The study addressed four main properties of learning objects: architecture (in terms of separation of data, logics, presentation and implementation of interaction interfaces), pedagogical contextualization, the use of standards, and the extent to which they are decomposable/composable.

As discussed in chapter 3.6.2, learning objects are frequently referred to as being reasonably context independent, freestanding, and platform neutral chunks of content that can be contextualized and assembled into larger units (learning modules) for use within specific pedagogical contexts. There is a conception of learning objects as a highly flexible and modular model for learning content that has the power to change the way in which teachers and producers approach digital learning content. The study presented in paper 1 shows that this conception is not entirely right and that there are many unsolved issues that need to be dealt with in order for learning objects to comply with the used definitions. Current practice of implementing and using learning objects, compared to commonly referred to definitions and taxonomies of learning objects, lead to the discovery that learning objects are often ascribed a number of properties and qualities that they do not automatically possess. There appears to be quite a discrepancy between how the learning object concept is presented and defined, and how it is practiced. This problem usually seems to be caused by the lack of

⁴³ The study was anonymized in order not to single out learning object producers as producing learning objects of poor technical quality.

criteria stating how learning objects properties should be implemented. The learning object concept appears to be too fuzzy and imprecise and in need of explicit, compulsory rules and best practice guidelines.

An issue that became clear in the study was that most learning object definitions and learning object taxonomies are too forgiving in that they allow too much to be included. An example of this is one of the most used definitions, the IEEE Learning Object definition (see chapter 3.6.2, page 24) that includes just about anything that is digital and can be used for learning. Another issue is that most of the studied learning object definitions stated a number of quite advanced properties and characteristics, without saying anything about how those can be accomplished in a conformed and standardized way.

The above issues appear to be associated with a frequent disregard of basic software architectural considerations, directly related to the lack of technical quality criteria and best practice. The major result of the study was the suggestion of six action areas for establishing technical quality criteria for learning objects:

- (1) There is a need for a common (more narrow) definition of what is, and what is not, a learning object. Excluding rather than including, which also means accepting trade-offs in order to gain a functioning concept by defining “must-have” properties and attributes. This includes the separation of pedagogical and technical issues as stated above. Experience from object-orientation and component technology should be considered.
- (2) In connection to narrowing down definitions, there is a need for a taxonomy that complies with the definition, and where granularities as well as special properties are regarded.
- (3) Standards used for learning objects should be extended to go beyond descriptive information, such as metadata, sequencing, and packaging to also embrace standards for interfaces, machine-readable descriptions of technical properties and interaction interfaces.
- (4) There is a need to establish standards and recommendations that address the internal use of data formats and data structure. This suggestion is related to area 3 and is important for the exchange of data as well as for separating data from logics and presentation.
- (5) To enhance the level of decomposability and context independence, it should be prescribed for the architecture of learning objects to be layered as a part of best practice, in order to separate data, presentation and application logics.
- (6) Pedagogy should preferably (when it is possible) be kept outside learning objects in order to facilitate pedagogical context independence. The contextualization (the pedagogical design) should be done when learning objects are composed into learning modules.

The results presented in paper 1 provide the basis for further addressing research question 1 and 2 by the suggestion of areas that need to be addressed in terms of technical quality. Such work is needed in order to accomplish the abstract model and technical infrastructure that are needed to prepare a common (and working) modular model and framework for learning objects and VLEs.

4.2 Results of paper 2

“Virtual Workspace Environment (VWE): A taxonomy and Service Oriented Architecture (SOA) framework for modularized Virtual Learning Environments (VLE) – Applying the Learning Object Concept to the VLE”

The research presented in paper 2 builds directly upon the research and results that were presented in paper 1. Paper 2 addresses research question 1 by suggesting an abstract model for modularity that includes both digital learning content – i.e., learning objects, and the VLE.

Learning objects were discussed in detail in chapter 3.6.1 and VLEs and LMSes were discussed in chapter 3.7.1.

The main research result presented in paper 2 was an abstract model for modularity, departing from the six areas for quality criteria presented in section 4.1. Figure 4-1 illustrates the model, which was named the “*VWE Learning Object Taxonomy*” – after the VWE project, within which the taxonomy was developed. The VWE project was described in chapter 2.1.2. The objective of the research was to address the issues that were raised and discussed in paper 1, while still maintaining the compatibility with existing learning object taxonomies, such as the atomic learning object taxonomy (see chapter 3.6.1).

The VWE Learning Object Taxonomy consists of two main object⁴⁴ types with different purposes, at different levels of aggregation (see Figure 4-1).

The two main object types are: simple learning objects and resource objects. Simple learning objects usually consist of a data object, or fundamental data object (FDO), which are equivalent to Wiley and Nelsons fundamental learning objects (FLO) (see chapter 3.6.1). FDOs are the most basic building blocks used for the aggregation of learning objects. FDOs usually consist of a picture, XML-data, an animation, data, etc.

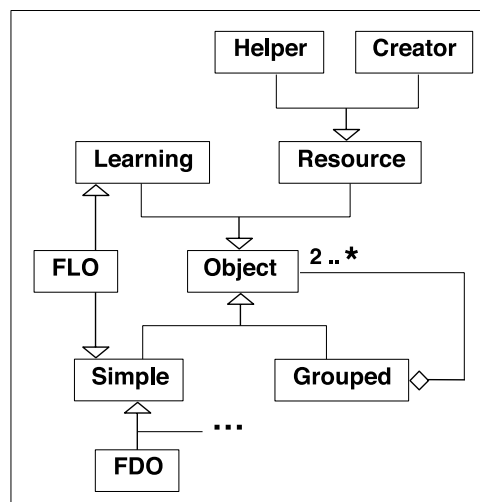


Figure 4-1. A concept map outlining the VWE Learning Object Taxonomy.

Resource objects are of two types: *helper resource objects* and *creator resource objects*. Resource objects add functionality in terms of application logic and by that they facilitate the separation of data and application logics. Helper resource objects are used to interpret, manipulate render and visualize data that (in the taxonomy) are simple data objects. The relation between helper resource objects and simple learning objects are similar to the relation between browser plug-ins, such as a viewer plug-in, and the data file to be viewed. Hence, the purpose of helper resource objects is to “help” other objects by providing suitable application logics. The purpose of creator resource objects is to add other application functionality and services to the VLE or to learning objects, other than functionality in the sense that helper resource objects provide. Creator resource objects, unlike helper resource objects, can also be used as stand-alone. They can implement any functionality, which means that they can be used as building blocks for modular VLEs in ways that are similar to software components.

⁴⁴ Note that the term “object” is not used in the same sense as an object in OOP.

However, creator resource objects can also be used in aggregations with helper resource objects and FDOs to form advanced, aggregated learning objects that are “packaged” together with VLE functionality.

As implied above, learning objects according to the VWE learning object taxonomy can be either singular or grouped/aggregated. Aggregations are called *grouped learning objects*, and grouped learning objects are the result of an aggregation of two or more objects. As such, they can be aggregated using any kind of objects from the taxonomy, including other grouped learning objects. An aggregation of grouped learning objects that is prepared for a specific learning situation is referred to as a *learning module*. It is at the learning module level that the pedagogically contextualization occurs. It should be emphasized that learning modules could constitute content as well as the VLE or parts of the VLE. Hence, according to the VWE Learning Object Taxonomy the VLE can be assembled and aggregated in the same fashion as learning objects are intended to be used, according to most definitions, when addressing digital learning content. Figure 4-2 and 4-3 are partly made out of screen dumps from the VWE VLE (see chapter 2.1.2), and illustrate how this works in practice. Figure 4-2 illustrates how a Simple Learning Object – in this case a file describing a chemical molecule, using the XML-based Chemical Markup Language (CML) – works in an aggregation with a Helper Resource Object, i.e., a CML viewer. Together they form a Grouped Learning Object that is pedagogically relatively context independent and that can be used to explore a chemical molecule. The chemical molecule is actually retrieved from an eternal database and loaded into the viewer. This means that the same viewer can be reused in a different pedagogical context involving chemical molecules, i.e., no hard coding of application logic.

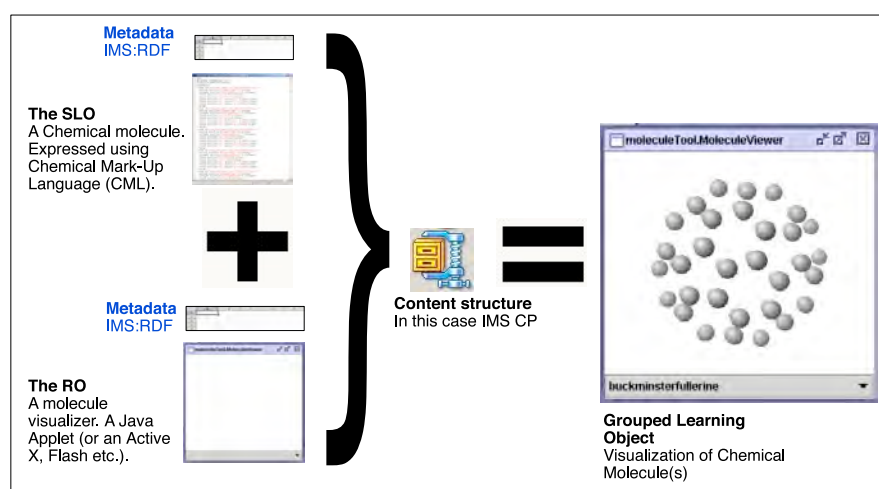


Figure 4-2. Illustrating the relation of a Simple Learning Object – in this case a CML/XML-file, and a helper resource object – in this case a CML Viewer. These are kept together using an IMS CP structure to form a unit that is a Grouped Learning Object.

The second example (Figure 4-3) shows how the same Grouped Learning Object is used in a new, and completely different, context. A knowledge test is put together and the molecule viewer and the CML molecule is aggregated together with a test question in the context (aggregation) of a knowledge test that (in the example) contains three questions. In this case, the test complies with the *IMS Questions and Test Interoperability* (IMS QTI) specification.

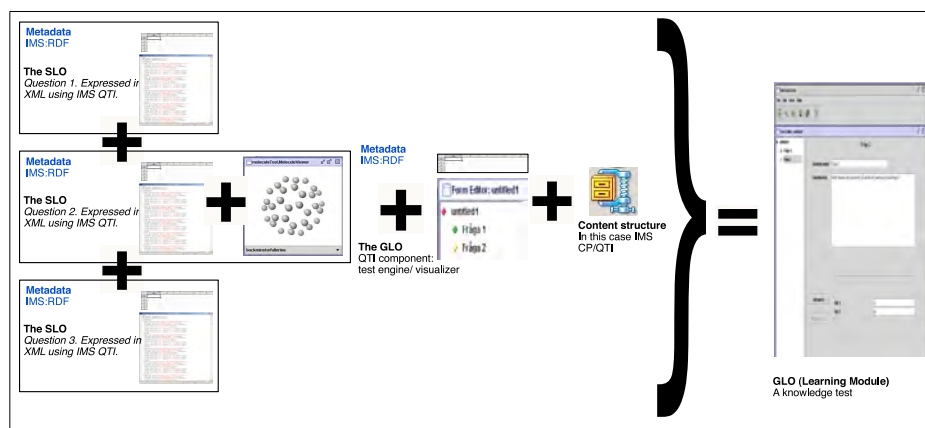


Figure 4-3. The relation between several VWE Objects; in this case test questions that are grouped learning objects assembled using a fundamental learning object and a helper resource object. The second question uses another grouped learning object to visualize chemical molecules. The result of the assembled and sequenced questions is a learning module – a knowledge test.

Except for the Grouped Learning Objects from Figure 4-2, the aggregations in Figure 4-3 also consist of two Resource Objects. These are Helper Resource Objects for dealing with IMS QTI test questions and IMS QTI tests and a Creator Resource Object that act as IMS QTI editor and that was used to create the test. The IMS QTI editor consists of a client and a server, exposed as a service.

4.3 Results of paper 3 and paper 4

“Suggesting a SOA-framework for modular virtual learning environments: comparing two implementation approaches” and “A Service Oriented Architecture-framework for modularized Virtual Learning Environments”

For clarity reasons, the results presented in papers 3 and 4 are summarized together. Each paper has a different focus, but the results are dependent on each other and the picture becomes complete only when the results are considered in relation to each other. The research presented in papers 3 and 4 is a consequence of requirements that arose from the results presented in papers 1 and 2. The papers address research question 2 by suggesting a modular, SOA framework that facilitate the implementation of modular VLEs and learning content in accordance with the VWE learning object taxonomy.

Paper 4 summarizes the final results of the system design and implementation of a modular architecture-framework, called the VWE architecture-framework, which complies with the VWE Learning Object Taxonomy, while paper 3 compares two experimental implementations, using different technology platforms for implementing the VWE architecture-framework. The implementations lead to prototypes of a Java-RMI based implementation, respectively a Web Service based implementation. The purpose of implementing the architecture framework twice, using different technology frameworks, was to provide a proof of concept, showing that the VWE architecture framework is general enough to be implemented using different technology platforms, and by that also general enough to provide the basis for suggesting it as a best practice framework approach.

The implementations resulted in two prototypes of web-based modular frameworks (called VWE I and VWE II) for the aggregation of VLEs and digital learning content, emanating from the concepts and ideas expressed by the VWE Learning Objects Taxonomy. A VWE VLE (including the content to be used) is assembled through the aggregation of components from the VWE Learning Object Taxonomy (see Figures 4-1, 4-2 and 4-3). The metaphors of *tools* and *workspaces* are used for addressing the components and the resulting VLE, respectively. Java applets were used in both prototypes for the implementation of application logics on the client, but other client-based technologies, such as flash, ActiveX or Ajax could also have been used, even though these were not tested.

The resulting VWE SOA framework is described in Figure 4-4. The SOA framework consists of five basic services; User Service, Message Service, Workspace Service, Tool Service, and File Service. Tools and other services in the VWE service registry can use these basic VWE services, in order to aggregate new services and VWE Tools.

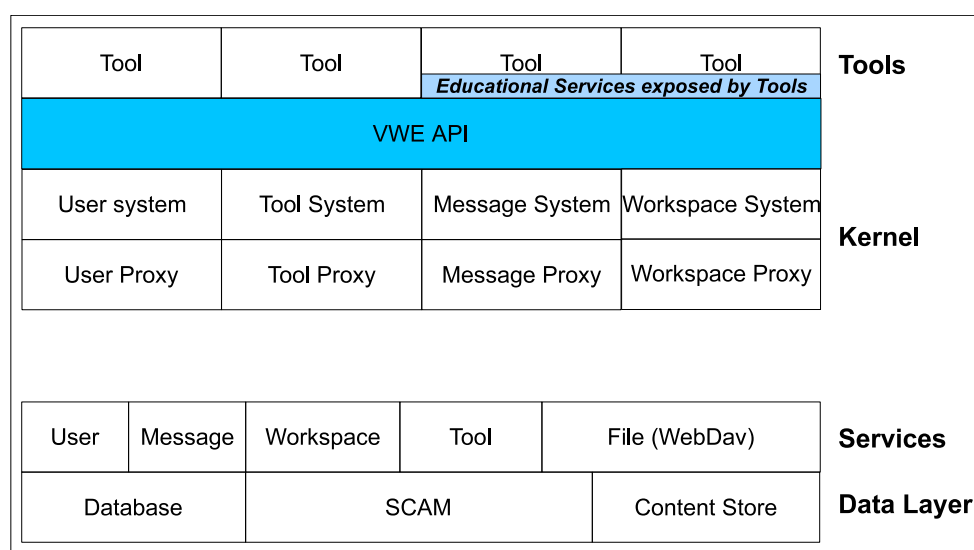


Figure 4-4. The VWE SOA architecture-framework, showing a set of common services that can be utilized by tools and other services.

Each service is implemented in accordance with a common pattern and design principle that consist of a Tool-System/proxy-Service. The same design principle is used in both prototypes.

The Web Service approach, which was found to be the most versatile and suitable for this kind of modular architecture-framework, uses UDDI (see chapter 3.11.3) for registry and discovery of services and tools. While UDDI was used as registry from the Web Service point of view, the SCAM RDF-store was used to manage educational metadata and some basic annotations of VWE-tools and their associated services. The SCAM RDF-store (the SCAM ePortfolio) was also used to manage user generated content and files together with user created metadata.

While the structure of a VWE workspaces was expressed using an IMS CP structure, it was implemented using a RDF-binding of IMS CP. The use of IMS CP allows workspaces to be represented and packaged in a standard way. However, IMS CP was not optimal for this purpose, which (together with a striving for consistency) was the reason for using an RDF-binding of IMS CP in order to create the potential to utilize the power of RDF and machine-processable semantics.

4.4 Results of paper 5

“Treating metadata as annotations: separating the content mark-up from the content”

Paper 5 addresses the development of an infrastructure and prototype (the Annofolio) for adding externally stored RDF annotations to XML documents. The resulting prototype is based on a methodology where metadata are regarded as being layered (illustrated in Figure 4-6) with layers of metadata markup that spans structural XML markup, semantically machine-processable metadata, and traditional bibliographic metadata (see chapter 3.9). By suggesting a methodology and infrastructure that can be used to enhance the representation of pedagogical instructions and processes through the addition of RDF-annotation and the possibility to use ontologies, paper 5 addresses research question 3.

The Annofolio methodology, which separates content, structural markup and semantic markup, and the corresponding prototype were the major research results presented in the paper. The Annofolio methodology facilitates machine processing and atomization of the processing of content (i.e., data), structure and metadata. The methodology makes it possible to apply machine-readable and machine-processable statements, not only to a resource as a whole, but also to a part of the internal structure of an XML resource, without altering the resource. This was made possible by storing metadata as RDF-annotations, separated from the resource using the Annotea W3C recommendation.

Figure 4-5 outlines a schematic view of the Annofolio methodology, as it was implemented as a prototype in the experiments that were described in papers 5 and 6.

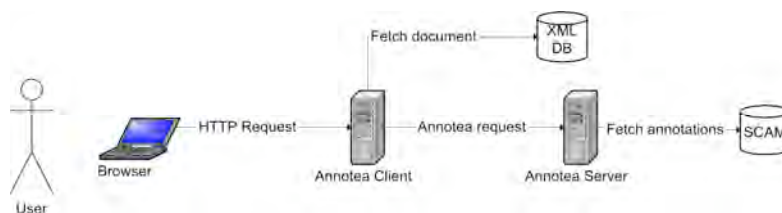


Figure 4-5. The principle of querying the system for an XML-document to browse and/or annotate.

XPointers are used to “point-in” annotations to a specific location in an XML file. The XPointers are specified based on the location of the annotation relative to a specific location of the documents XML *Document Object Model* (DOM) – that is, the number of characters relative to the nearest tag in the DOM tree. The Annotea compliant annotations are stored and managed in a SCAM RDF store that is working as an Annotea server. The XML resources are stored in an Apache Xindice XML database.

The use of Annotea and RDF makes it possible to express any kind of metadata and semantics, according to any application profile or metadata model. The Annofolio method makes it possible to attach any number of metadata records to a resource, or to different parts of the internals of a resource. These metadata records are regarded as different layers of metadata and they are managed completely independently of each other and different users can manage different layers. The annotations can be stored in any Annotea compliant RDF-store, which means that several independent metadata stores can be set up and maintained by different independent actors in a metadata ecology (see chapter 3.9), for example, by utilizing an existing Annotea infrastructure together with the Annofolio. The Annofolio metadata ecology is illustrated in Figure 4-6.

4.5 Results of paper 6

“Marking the National Curriculum - a new model for semantic mark-up”

The research presented in paper 6 builds upon the results presented in paper 5. The study presented in paper 6 examines how the Annofolio prototype can be used to add metadata annotations to the *Swedish national curriculum*. The use of the Annofolio prototype was compared to a reference implementation using a traditional approach to content markup, based on TEI, where the semantic markup is tagged inside the XML resource using XML tags from the TEI schema. This means that the resource has to be altered in order to add metadata and that structure metadata and metadata semantics are mixed – a practice that was discussed in chapter 3.9.4. Research question 3 was addressed in paper 6 by showing how RDF-annotations can be used to dynamically connect different components of TEL, by utilizing metadata semantics.

The Annofolio prototype was tested against five criteria provided by the *Swedish National Agency for Education*⁴⁵ for the use of the *Swedish national steering documents*,⁴⁶ together with the needs provided by the objective of adding metadata markup to the national curricula. These requirements were: (1) The content, semantics and meaning of the original document must not be altered; (2) The format for data distribution must be structured, open and application neutral; (3) Markup of a whole resource, as well as of a part of a resource, must be supported; (4) Several application profiles and metadata models must be supported for the same digital resource, as well as for parts of a resource; (5) It must be possible to add markup without interfering with the original document or existing metadata.

The Annofolio and the associated methodology managed to fulfil all five criteria while TEI prohibits metadata markup by a third party, which means that metadata may only be added by the controller of the resource. TEI also forces alteration of the original resource in order to add metadata markup. Approaches like TEI, where metadata and semantics are incorporated in the XML file together with syntax and structure, also prohibit the use of more than one application profile or metadata model at a time, hence prohibit the principle of separating syntax and semantics.

In theory, the Annofolio methodology allows anyone to add metadata without altering the resource or interfering with existing or future metadata. The experiment also shows that it is possible to use several different metadata models for the same resource and at the same time. This contributes to the creation of a metadata ecology – according to the principle that is illustrated in Figure 4-6.

⁴⁵ Skolverket (<http://www.skolverket.se/>), which is the Swedish agency responsible for the national curriculum.

⁴⁶ The Swedish steering documents are a collection of documents for Swedish schools that contain the curriculum, course plans, laws and regulation etc. Skolverket is the agency responsible for the administration of the Swedish national steering documents.

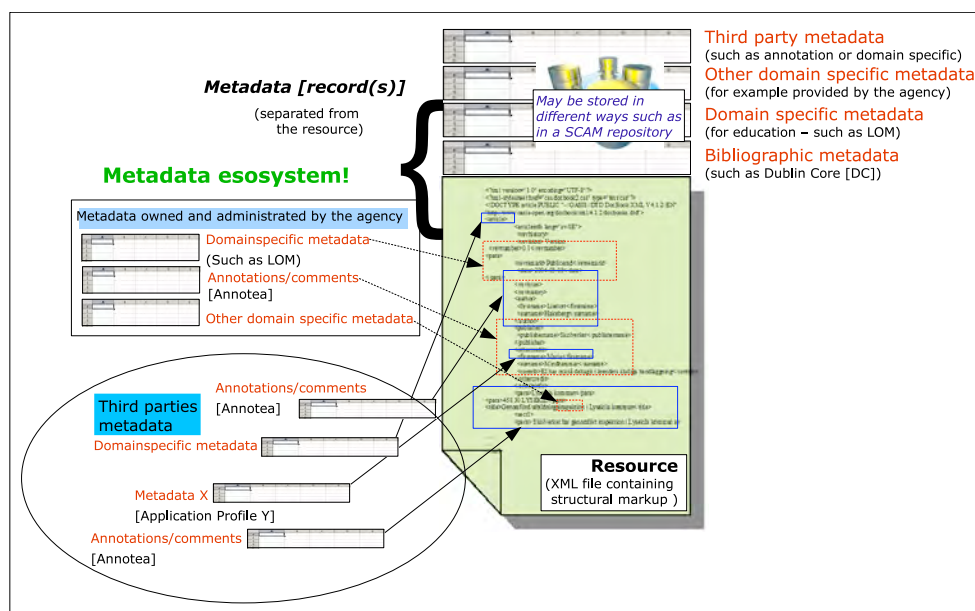


Figure 4-6. The principle of an Annofolio metadata ecology where metadata is “pointed in” as annotations to the XML resource, or to a specific location within the DOM-structure of the XML resource.

A small and well-defined experiment showed that the markup method used by the Annofolio could also be used to map the annotated curricula to digital learning resources that were annotated with corresponding metadata. Mapping between curriculum and digital learning resources was tested using annotations in the curriculum as input for the search and filtering of digital learning resource metadata in a LOR. The results of the experiment showed that the Annofolio could be used to create dynamic relations between the curriculum and learning resources based on RDF-annotations. Such an approach makes it possible to locate digital learning resources that are relevant in a specific pedagogical context, which would not have been found using a traditional search methodology. As an example, digital learning resources that were relevant for one subject (history) were, for example, found in a repository containing digital learning resources intended for a completely different subject (chemistry). In practice, this phenomenon occurred when annotations to the curriculum in history were used as input to form a query to the LOR, and the returned result set suggested learning resources intended for chemistry about how soap was made in the Middle Ages.

The prototype was also subjected to another experiment where it was tested for the compilation of “digital compendiums”. Adding annotations to the curricula documents marked off the scopes of the text excerpts to be included in a compendium. Compendia were generated using XSL transformation to transform the XML excerpts from the original documents into a new XML file. The new XML file was used to generate a number of different output formats. The experiment indicated that the use of annotated curriculum could become versatile and a useful component for TEL, and as a part of the metadata ecology.

The results that have been presented in this chapter will be discussed in detail in the next chapter, where I will also discuss how the presented research relates to each other and also suggest some ideas and strategies for further exploration.

5 Discussion and conclusions

This chapter contains a supplementary discussion and analysis of the results and conclusions of the research presented in the appended papers, as well as of their relationship to the theoretical framework that was presented in chapter 3. The results are discussed in chronological order as they appear in the papers, ending with a summarizing and concluding discussion followed by suggestions for future research directions. The reader is referred to the appended papers for a more detailed description and discussion of the research results.

5.1 Introduction

This thesis attempts to contribute to the understanding of the relationship between pedagogy and technology, especially contributing to the understanding of the role of modularity in relation to pedagogical flexibility and the ability of technologies to adapt to different pedagogical approaches and design of pedagogical activities. This ability (discussed in chapter 3.2) has become increasingly important as the expected learning objectives and learning outcomes are continuously changing and often consist of tasks requiring complex cognitive processing. It is indicated by the theoretical framework (chapter 3) that different types of knowledge leads to the formulation of different learning objectives that require different levels of cognitive processing, leading to a need for different pedagogical approaches that are associated to different learning theories. Hence, flexibility from both a pedagogical and technological perspective is becoming an increasingly important characteristic of learning technology and digital learning content.

The theoretical framework for this thesis, as presented in chapters 3.4, 3.6, and 3.11, indicates that modularity plays an essential role in contributing to flexibility, especially in terms of adaptability and adaptivity. This has also been proven by modular approaches within other areas, such as within computer science (see, e.g., Nierstrasz & Dami, 1995; Szyperski, 2002b; Weinreich & Sametinger, 2001). See also chapter 3.11.

The purpose of the studies presented in this thesis has been to examine how, and to which extent, modularity can improve pedagogical flexibility, and to show how experience from computer science can be applied for this purpose (see chapter 1.4).

Three research questions were formulated for this purpose:

- 1. How can a common abstract model for modularity, which covers both the VLE and digital learning content, be accomplished without creating incompatibilities with existing models?*
- 2. How can a technical infrastructure be designed and implemented according to a common abstract model, in order to provide the necessary supporting infrastructure, and without disqualifying legacy?*
- 3. How can pedagogical designs and processes be represented in order to utilize modularity to increase pedagogical flexibility and adaptability?*

In the following chapters I will discuss the results presented in the appended papers in relation to those research questions. I will also discuss some of the strengths and weaknesses of my results in relation to the theoretical framework and to other studies within this research area. In the final part of this chapter I will suggest some future research and theoretical approaches that seem appropriate in order to take the research presented in this thesis to the next level.

5.2 Learning objects and modularity

The concept of learning objects (discussed in chapter 3.6) is likely to be one of the most hyped concepts within TEL during the last 10 years. In spite of the high expectations on learning objects and large economic investments, the success has so far failed to appear. Learning objects are often regarded as a way of enhancing reuse and interoperability (see, e.g., Polsani, 2003), and thus, lowering the production cost of digital learning resources by submitting them to a standard approach that allows the combination of learning objects (from different vendors) into aggregated units to use for learning in new contexts. In business terms this is sometimes referred to as a *learning object economy*, which has been predicted to reach a multi-million Euro turnover, which in turn is expected to catalyze the development of learning objects and by that also favor the pedagogical upsides of learning objects (see, e.g., Hylén, 2002; Polsani, 2003; ADL, 2004). However, the learning object economy is still waiting to happen.

The study presented in paper 1 shows that several important reasons for these problems lie at the technical implementation level. The study shows on the one hand that there is quite a wide gap between the definitions of and expectations on learning objects, expectations that are often based on functional requirements, and to some extent even a spoonful of wishful thinking, and on the other hand the way that learning objects are implemented. This can be compared to the situation within computer science and component-orientation, where a similar situation was experienced years ago (see chapter 3.11). The problems that arose within component-orientation were partly addressed by regarding component properties from two different viewpoints: technical properties and methodological properties also referred to as functional properties (cf. Polsani, 2003) or business properties, depending on the adopted vocabulary. By making such a distinction it is possible to separate technical and pedagogical (functional) aspects of learning objects, while still keeping them both in view.

My study shows that most learning object definitions are far too wide to be realistic because they include digital resources that range from anything to everything. Similar conclusions can be found in the work by Wiley (2002a), Polsani (2003), and many others. From a content perspective ‘from anything to everything’ might be a good quality (excluding nothing), but from an implementation perspective it can be devastating. One of the most significant effects indicated by the study is that most learning objects can only be *packaged and sequenced* together, but they do not actually *function* together, and they are unable to support and utilize each other in the way that is possible with software components in composite applications. Neither can they be subordinated to a common structure, nor a common look-and-feel, because application logics, data and presentation are often mixed, and thereby locked to the context of that specific learning object.

In paper 1, the technical properties were addressed from the point of view of technical quality. Technical quality is a necessity in order to address research question 1, and by that a necessity in order to address research question 2 as well. Without distinct and unambiguous technical definitions and properties there will be no *common abstract model for modularity* and no *technical infrastructure will be designed and implemented according to a common abstract model*, which are the core values and objectives of the research presented that addresses research question 1 and research question 2.

However, in reality this kind of technical quality work cannot be done in isolation and presented in one research paper. Such work has to be subjected to a process of standardization and consensus, preferably verified by working implementations as a proof of concepts (such as the prototypes discussed in paper 3 and paper 4), and not only through ‘standardization by committee’. This is also the reason why six areas for further action were suggested instead of

explicit quality criteria. Such work needs to be carried out in a broader context. However, the suggested areas for quality criteria are good enough to use as temporary criteria in order to continue the work presented in paper 2, paper 3 and paper 4, which provide a proof of concept of the chosen approach, addressing research question 1 and 2 by defining and implementing a learning object taxonomy and a corresponding modular architecture framework.

One of the reasons that the concept of technical quality was chosen as a starting point is that this approach matches previous research on learning object quality frameworks done by others, such as by Van Assche & Vuorikari (2006). It is important that the work on technical quality can be incorporated in existing and future work in order to be adopted and gain momentum.

5.2.1 Reviewing the technical quality criteria

The six suggested areas for further work on technical quality (see chapter 4.1) are briefly discussed below. They should be seen as suggested points of departure for further work in terms of research, standardization and consensus building. The reader is referred to paper 1 for additional details.

The first suggested area – a common narrow definition, defining “must-have” technical properties and attributes, while separating technical and pedagogical properties (as discussed above), is most likely essential for the survival of learning objects as a concept. This is also indicated from the development of component technology that was discussed in chapter 3.11. It is also essential for the work in addressing the other five suggested areas.

The second suggested area is closely related to the first area, but should be a separate area, as a taxonomy, like the one discussed as a result of the research in presented in paper 2 (see chapter 4.2), can be regarded as an refinement of a working definition that can be implemented, while still being an abstract model. Such a taxonomy model is important as it formalizes the definition by the use of clear, unambiguous semantics that ultimately either includes or excludes properties in a way that is comparable to the UML modeling of computer systems. The VWE Learning Object Taxonomy in Figure 4-1 illustrates this. This type of “forced clarity” is promoted by the research of Song and Andersson (1999), where they take on a rather strict object view similar to the object definition within object-orientation. However, the objects within VWE Learning Object Taxonomy actually have more in common with software components than software objects, a discussion that I will return to later in this chapter.

The third suggested area addresses technology standards. Technology standards are essential in order to realize modularity (see chapter 3.6.1). The development of learning technology standards needs to be based on general technology standards, such as the standards for SOA and Semantic Web Services, discussed in chapter 3.11.2 to 3.11.4. This is indicated by the research presented in papers 3 and 4. Such development is also being done in projects like the e-framework (B Olivier, Roberts, & Blinco, 2005) and within *IMS General Web Services (IMS GWS)* (IMS, 2005a) (also discussed in chapter 3.6.1), but there is still a long way to go, and the research still needs to prove its validity through working implementations. While it is easy to see that this rather abstract level of standardization of learning technology needs to be a prioritized area for technical quality, it is in reality a long and quite arduous process that contains many pitfalls and obstacles that need to be dealt with. While standards are essential for the development of global systems such as the web, Web Services, learning objects etc. (see chapter 3.6 and chapter 3.8), there is always an intricate balance in finding the right level for standardization in order to make standards work in practice. These are issues that are

discussed by Duval (2004), where the standards process is outlined and this intricate balance is discussed, resulting in similar conclusions.

The fourth suggested area addresses the internal use (inside learning objects) of technology standard specifications. This is an important issue as the decomposability of learning objects depends on the fact that the resulting parts of decomposition must be usable in new contexts, i.e., they need to be standard-compliant. Considering the number of available competing and complementing content standards, such as for images or data formats based on different XML Schemas, there is a need for recommendations and best practice for use when authoring and developing learning objects.

The fifth suggested area addresses the software architecture of learning objects. The software architecture of learning objects needed to be considered when addressing research question 2. The research presented in paper 1, and especially the suggestion of area four, is a first step toward addressing the software architecture aspects of learning objects.

One matter that was evident from the study is that the software architecture aspects of learning objects are in general not considered, and many of the learning objects in the study could actually be considered to be black boxes *without interfaces*, which can be compared to Szyperski's (2002a) classification, which was also discussed in chapter 3.11. A black box component without interfaces is virtually useless from a composite perspective. There is no way that such learning objects can be aggregated in a way that make them *function together* with other learning objects. They cannot pass messages, they cannot receive messages, they cannot be subordinated to the same look-and-feel, and they cannot be decomposed. The exceptions in the study were some of the SCORM-compliant learning objects that were able to interact, but only within a SCORM environment. The SCORM reference framework was briefly discussed in chapters 3.6, 3.7, and 3.10. SCORM compliance is of course better than the first case, but it is still problematic, because SCORM does not by necessity make learning objects decomposable. A second problem is that SCORM is only supported within the learning community and it is merely a reference model for learning objects and LMSs that does not only rely on standard specifications, and by that it becomes a somewhat proprietary and narrow framework (see, e.g., Downes, 2002; Friesen, 2004b), which may even add to the problem (chapter 3.7.1). In addition, the need for components of the VLE to communicate and interact with systems and services outside their own system borders is constantly increasing. This is indicated by projects such as the e-framework, together with the occurrence of related concepts and ideas such as *Managed Learning Environments* (MLEs) (JISC, 2003). The implementation of a completely modular, service-based, and composite environment is also discussed related to the research results presented in papers 3 and 4, and that discussion makes this standpoint even clearer. It may even be legitimate to question whether a concept like system borders is even applicable to the needs of such environments?

The issues described above provide strong arguments for using general technology standards as the basis for learning technology standards and the implementation of learning technology.

The sixth suggested area addresses the representation of pedagogical instruction (i.e., instructional design and learning design). A concrete suggestion was made in paper 1 to separate the representation of the pedagogical design from single learning objects. One consequence of this is that pedagogical instruction should not be hard-coded into learning objects, which was the case with most of the learning objects in the study. Pedagogical processes should be implemented in a separate layer, using a standardized data format and meta-languages for such representation (see chapter 3.10). There are currently a number of available options, such as IMS Simple Sequencing, SCORM Sequencing and Navigation and IMS LD. However, these standards are still rather immature and are all inflicted with their

respective problems and limitations. The separation of pedagogical instructions and learning object internals is important for enhancing decomposability as well as composability in terms of pedagogical context-independence. The suggested separation is of course not intended as an absolute rule, rather as a suggestion for best practice where it is possible – ultimately it will be a matter of design decisions based on common sense, design patterns and experience.

Meta-languages and data formats for representing pedagogical instruction can be used in several ways, such as internally within a learning object as an alternative to hard-coding, or as a “glue” that binds several learning objects together into learning modules by representing interrelations in terms of a pedagogical context and design. The latter was exemplified in paper 3 (see Figure 4-2 and 4-3) using IMS CP that is rather unsophisticated and mainly designed for packaging and structure, but the principle of separation is the same.

The relations between learning objects and pedagogical contexts are a debated issue, and there are several views of how this issue should be handled. For example, Rehak & Mason (2003) suggests that learning objects need to have a built-in pedagogical purpose by necessity, otherwise they would not be learning objects, while McCormick (2003) argues that pedagogy should be kept out of learning objects for the reason of pedagogical neutrality and reusability. The hypothesis brought forward in this thesis is that pedagogical instruction must be technically (i.e., software architecture-wise) separated, at least into its own layer, and that pedagogical context should be created at composition time, using EMLs.

The suggested work areas for establishing technical quality criteria were addressed by the research presented in papers 2-5, through the implementation of a joint learning object taxonomy for digital learning content and the VLE – the VWE Learning Object Taxonomy, which will be discussed in the following chapters.

5.2.2 Suggesting an abstract model for modularity

The main result of the research presented in paper 2 is the VWE Learning Object Taxonomy, briefly described in chapter 4-2. The VWE Learning Object Taxonomy is one of the most important results presented in this thesis, as it constitutes the axis around which much of the other research presented here revolves.

The objective of the VWE Learning Object Taxonomy is twofold: to address the issues regarding the architecture of learning objects that were discussed above, and to outline a model for modularity that is common for both digital learning content and for the VLE. The purpose of the latter model is to apply the modular concept of learning objects to the VLE as well. As discussed in paper 2 and in chapter 3, there is a mismatch between the modular concept of learning objects and the non-modular concept of the LMS. An LMS merely becomes a container for learning objects, complemented with tools for administration of learning and learning content, while many pedagogical aspects are set aside or excluded in order to be possible to implement by learning objects. This creates additional issues regarding where pedagogical instruction should reside – referring to area 6 in paper 1.

Even though pedagogical instruction can (and in some cases must) be implemented inside learning objects or at the learning module level (implemented as grouped learning objects according to the VWE Learning Object Taxonomy), there is still a need for applications and pedagogical tools to be includable as modules in the structure, because this is a natural consequence of the design of pedagogical instructions that utilizes learning objects. However, this cannot be easily managed as long as the needed tools are locked into the proprietary silo-like system that most LMSes provide (see 3.7). In addition, it is very hard to establish where the responsibility of the LMS ends and the responsibility of learning objects begin in terms of providing application logics. This problem does not occur in a modular environment because

the VLE - or at least the instance of a VLE generated for a specific context - and its content does not actually exist until someone composes it, which is also the major idea of learning objects. In the same sense that Nierstrasz & Dami (1995) claim that there is a distinction between *composing-time* and *run-time* in component-based software (see 3.11), there is a similar distinction in a modular VLE. A modular VLE can actually be regarded as a type of composite application. However, this distinction does not mean that the composition cannot change after composing-time. There is an initial context that may change during run-time, which in turn may trigger changes in the composition by actors in the system. One technical factor that promotes this behavior is the relative lack of hard bindings between different components, which is a consequence of loose coupling. This characteristic can be compared with object-orientation, where it does not exist because there is no loose coupling.

The immediate effect of applying the VWE Learning Object Taxonomy is that it becomes possible to apply a common modular model for both the VLE and the digital learning content, which also answers my research question 1. At the same time the VWE Learning Object Taxonomy addresses several of the architectural issues discussed in paper 1.

The VWE Learning Object Taxonomy is compatible with most of the other learning object taxonomies and definitions that were discussed in chapter 3.6.2 (as well as several of the definitions that were not discussed), and should therefore be regarded as a supplementary taxonomy rather than a replacement. The reason for this is that the use of the VWE Learning Object Taxonomy together with taxonomies addressing pedagogical and complexity aspects of learning objects, such as Wiley's (2002a) Atomic Taxonomy (discussed in chapter 3.6.2), makes it possible to adopt a distinction between *methodological properties* and *technical properties*, similar to the distinctions made by Nierstrasz & Dami (1995), as well as by others, from the point of view of Component-Oriented. Song and Andersson (1999) draw similar conclusions, even though they refer to more traditional properties of Object-Oriented rather than drawing parallels to Component-Oriented. However, most existing learning object taxonomies and definitions contradict this view as they do not address technological properties (see, e.g., Wiley, 2002a; McGreal, 2004; Polsani, 2003). The above distinction of properties could be a good viewpoint to start addressing some of the confusion of ideas that were discussed in chapter 3.6.2 and it may also help in bringing pedagogy and technology closer together without disqualifying one or the other.

Beside the technical argumentation, there are strong pedagogical reasons to have a joint modular model as well. The fact that there are different pedagogical approaches, methods and learning theories, which are applicable for different pedagogical requirements, put flexibility in the front row. One of the main arguments for modularity is flexibility – technical as well as pedagogical. The emphasis on flexibility is evident from the development of modular approaches within computer science, such as Object-Oriented, Component-Oriented, and Service-Oriented (see chapter 3.11), as well as from the pedagogical field, such as chunking and sequencing (see chapter 3.4 and 3.5), and learning objects (see chapter 3.6.2).

As an example, Ertmer and Newby (1993) describe learning theories as a continuum, where different learning theories are suitable for different types of knowledge and by that also suit different pedagogical approaches (see chapter 3.2.4 and chapter 3.3). The situation is complicated by the fact that one learning situation may include different learning objectives, involving different types of knowledge, leading to several methodological approaches. The effect of this is a need for highly flexible and adaptable VLEs.

The research presented in the appended papers addressed this need through modularity, and by the implementation of the VWE Learning Object Taxonomy based on the SOA-based VWE Architecture Framework. The VWE Architecture Framework (see papers 3 and 4)

illustrates how the VWE Learning Object Taxonomy can be translated into a software architecture framework, and the implementation of the two prototypes proves that it is possible to implement a modular framework that includes VLEs and digital learning content into the same conceptual model, and by that extends the learning object concept to embrace the VLE. Altogether, this creates the potential to achieve the flexibility and adaptability needed in order to address different pedagogical needs and approaches, i.e., addressing research question 2.

5.2.3 From abstract model to architecture framework

While the VWE Learning Object Taxonomy appeared to be a practicable and well-founded approach to address research question 1, as well as the software architectural problems discussed in paper 1 (and above, in chapter 5.2.1) it needed to be validated and confirmed. The most reliable method to validate the taxonomy appeared to be through experimental implementation of the taxonomy, i.e., translating the VWE Learning Object Taxonomy into an architectural framework (see Figure 4-4) that could then be used as a blueprint for implementation.

The VWE Learning Object Taxonomy mainly addresses technical properties of learning objects from a client perspective or from a stand-alone application perspective. However, as the taxonomy is supposed to embrace digital learning content as well as the VLE, there is a need for a corresponding architecture on the server. Even though the functionality that is addressed by the core of the architecture framework is functionality that is traditionally implemented by the VLE, it is often useful for learning objects to be able to access and utilize those services independently of the VLE, mainly for reusing and sharing application logics and in order to avoid redundancy.

When designing an architecture framework, such as the one described in Figure 4-4, there is always an issue of what level to choose in terms of the scope of the offered services. Similar considerations are also experienced with component-orientation (cf. Szyperski, 2002c), and service-orientation (cf. Erl, 2007). There are basically two ways to go. One way is to offer a smaller, but useful set of common services that are always needed in VLEs, which was the path that was originally taken by OKI for the development of the OSIDs (see, e.g., Thorne & Kahn, 2003). The other alternative is to extend the basic set of services, offering services that operate on a detailed pedagogical, methodological and administrative level, which is a strategy that was adopted for the e-framework (formerly known as the e-learning framework) (see, e.g., Olivier, Roberts, & Blinco, 2005; (Wilson, Blinco, & Rehak, 2004). E-framework and OKI OSIDs were discussed in chapter 3.6. It should, however, be noted that these frameworks are somewhat difficult to refer to, because they are moving targets and constantly changing. This is common for work in progress, which both OKI OSIDs and the JISC frameworks can be considered to be.

For the VWE architecture framework, a choice was made to focus on a few essential services that are needed in all VLEs. This choice was made for mainly two reasons. The first reason was very practical and concerned resources. Each additional service to be implemented is very resource-consuming, and the VWE project (see chapter 2.1.2) had limited resources. In terms of proving the validity of the VWE Learning Object Taxonomy, it would not have contributed much to the research to implement a large number of additional services, even though it could have been a pedagogical finesse in terms of communicating the results to pedagogues to provide even clearer examples of functionality. The second reason for choosing to implement only a few central services was that a larger number of pedagogical services create a risk of steering the pedagogical approach by offering a pre-selected choice of tools and services. This is, for example, the case with the JISC framework, in which it is

possible to discern traces of the British national curriculum from the selection and design of services and tools. Such influences are of course the effect of cultural and political differences between and within educational systems as well as the result of different pedagogical point-of-views. They are the outcome of different learning theories (see chapter 3.2) and different learning objectives (see chapter 3.3), which, in turn, influence the chosen instructional strategies (see chapter 3.4), and so on. For those reasons, keeping the architecture framework on a strictly technical and infrastructural level of abstraction may better preserve pedagogical neutrality.

New viewpoints occurred during the work with the two prototypes, which make it reasonable to discuss the above reasoning a bit further. It could for example be argued that the services in the VWE Architecture Framework should be a part of a general architecture and infrastructure instead. This is probably true, even though most organizations have no such general infrastructure – at least not yet. In cases where there are already existing services, such as identity management, identity federation, user catalogue (such as LDAP), Single Sign-On (SSO) and likewise, these existing services can be pooled and used to form the *VWE User.Service*, or parts of it. The services outlined in the architecture framework are schematic, and in that sense abstract, and some of them have already been implemented as several pooled services. This is a design choice that was made in order to increase flexibility (the pooled services are decomposable), and to decrease the level of complexity for VWE Tool developers that utilizes the services and their interfaces for the composition of new tools. This makes it possible for third party developers to work against a smaller number of interfaces, while maintaining the possibility to use parts of an existing infrastructure, while still maintaining the VWE service interfaces.

However, the design issues discussed above do not actually affect the relevance of the architecture framework as a blueprint for the implementation of the VWE Learning Object Taxonomy, because the basic principle of modularity remains. On the contrary, it actually strengthens the concept, because reuse can be transferred to the next level by using a general infrastructure – when the general infrastructure is ready to be used.

The core idea of the architecture framework is that it should provide a basic infrastructure that can be utilized when adding new service-tool pairs that target specific pedagogical needs, or other needs. This model makes it possible to implement other frameworks, such as the e-framework, on top of the VWE Architecture Framework. This way, the VWE Architecture Framework can be extended to embrace both additional frameworks and arbitrary tools in accordance with the VWE Learning Object Taxonomy.

The results of the experimental implementations of the architecture framework, resulting in two prototypes (VWE I and VWE II), are discussed in the next chapters.

5.2.4 Prototype implementations

As discussed in papers 3 and 4, SOA is a suitable approach for implementing modular and distributed software architectures, a conclusion that is also supported by others (see, e.g., Szyperski, 2002b; Sim, Wang, Gilbert, & Wills, 2005; Makola et al., 2006). This is also the reason why the Service-Oriented design paradigm (Erl, 2007) and SOA was used as a basis for the blueprint that is provided by the VWE Architecture Framework (see Figure 3-9).

SOA has a number of specific characteristics that make it suitable. There actually appears to be no realistic alternatives to SOA, because the architecture framework must function in a distributed fashion on the Internet and in a web environment. Services are implemented as freestanding units (functionality modules) that can be distributed over a global network, provided as encapsulated and *loosely coupled* software components. Loose coupling is one of

the important characteristics that make SOA so well-suitable, because it prevents strong dependencies between components (see chapter 3.11.2).

The concept of loose coupling is important for achieving modularity, because it allows services to be pooled together to form new functionality in new contexts – such as in the context of a VLE or to support digital learning content, i.e., as the server part of a *helper resource object*, (see Figure 4-1) that supports a specific pedagogical context. In the experimental prototypes, such pooling is accomplished by combining different *resource objects* to form *grouped objects*, which in turn are grouped into the components that are called *VWE Tools*. These are grouped into one or more VWE Workspace(s), which is an aggregation of grouped objects forming a VLE that contains personal and shared learning spaces, containing pedagogical (and other) tools. Each component utilizes a pool consisting of one or more services from the VWE service framework (see Figure 4-2) and, when applicable, tool-specific services that are implemented as the server part of a tool, provided by a tool-specific or third party server, which in turn may very well utilize other services within or outside the VWE Architecture Framework. The reader is referred to papers 3 and 4 for additional details on the concepts of VWE Tools and Workspaces.

5.2.4.1 Considering some common design issues

The principle of hiding the implementation details of services from service consumers makes the implementation details of the prototypes less important. The main focus of the discussion in this chapter will therefore be on design principles rather than on implementation details. Even though some implementation details are discussed in paper 3 and paper 4, these were mainly used as examples in order to illustrate common principles. The main concern is that the prototype implementations show that the architecture framework is possible to implement using a SOA approach – irrespective of technology framework. Taken to its extreme, even a poor implementation would have proven this – as long as it provides the needed service interfaces that hide it. Hence, the two prototypes verified that the VWE Learning Object Taxonomy is valid; i.e., it can be implemented, and that it also addresses the architectural issues outlined in paper 1. As a result, a conceptual space that is common for both the VLE and digital learning content has been created within the demarcations of the learning object concept, which means that research question 1 and 2 have been affirmatively addressed.

During the time-period of carrying out the research, presented in papers 3 and 4, Web Services became more or less de-facto, and the SOA concept is sometimes even used synonymously with Web Services. It is important though to emphasize that this is not the understanding put forward by this thesis. The use of two different SOA approaches for the implementation of the prototypes was a manifestation of this view and additional alternatives are discussed later in this chapter. Hence, for the research presented in this thesis, the distinction between Service-orientation, SOA and Web Services is important, a view that is also argued by others (see, e.g., Erl, 2007; Sim et al., 2005).

As discussed in paper 4, the prototypes were implemented using one approach based on Java-RMI (VWE I) and another approach based on Web Service technology (VWE II). Additional details on the respective implementations are given in papers 3 and 4, while the results of the research are discussed below.

The blueprint for both implementations was the VWE Architecture Framework, illustrated in Figure 4-4. Some minor adjustments were made for the Web Service implementation in order to solve some design issues related to the change of technology framework. It was estimated that these adjustments did not influence the results of the study in any important ways because the basic principals remained the same.

While the server technology frameworks differed between the two prototypes, the clients were implemented as Java Applets in both cases. This resulted in that the same client instances could be used by both prototypes, with the exception of minor revisions, such as bug fixes and small enhancements. The choice of Java Applets can be discussed, and the main reason for the choice of Java Applets was the practicality to avoid additional technology load brought on by the use of additional client technologies – considering that Java development was already being done. Furthermore, there was an impending risk of increasing the complexity, adding incompatibility problems by mixing different technologies and the objective of the research was not solely to address such issues. However, this also meant that there was no testing involving alternative technologies on the client, such as clients developed using Flash or Ajax. Even though alternative clients should have worked (in theory) in the second prototype, given the nature of Web Services, they would have been less likely to work in the first prototype, considering the nature of Java-RMI. Another drawback of the Java Applet approach is that it demands a browser plug-in, which may cause compatibility problems. The use of a browser plug-in also puts an extra load on users, a problem that must not be underestimated when working with education and learning. Such issues speak in favor of technologies such as Ajax as the preferred client technology. Even though Ajax is still a young and rather untested concept that has not yet been consolidated and still is afflicted with a lot of problems, it is a promising approach to rich web clients (see chapter 3.8.3).

Papers 3 and 4 provide additional details on the implementation of the services in the service framework. However, three of the services (the Tool Service, the Workspace Service, and the Message Service) are less straight forward and merit some additional attention in this discussion, mainly due to their central roles in enabling modularity by keeping things together by acting as a “glue”.

The Tool Service is responsible for locating, loading, running, managing, and deploying tools and content. Every component in the system has its associated metadata, which means that the Tool Service manages extensive amounts of metadata. The mutual allocation of tasks between the Tool Service and the Workspace Service could be discussed, referring back to the previous discussion about the granularity of services. The responsibility for the interaction of tools could, for example, have been allocated to the Tool Service instead.

The Workspace Service is responsible for keeping workspaces together by managing the creation of workspaces, associating tools and workspaces, and associating workspaces with users and groups, loading the right workspaces and tools at log-in, and so on. The Workspace Service is also responsible for managing the interaction between tools and workspaces, while the Message Service is responsible for passing messages between tools, or between two instances of a tool, i.e., the sharing of data by two actors in the system. One such example is the sharing of data between two or more users working with a collaborative whiteboard, which was one of the example tools, bundled with both prototypes.

The Workspace Service handles the structure and packaging of workspaces. This is done using IMS CP, or rather an IMS CP Binding to RDF. The main reason for the choice IMS CP was that it (at the time) was the only available standard specification for representing package structures for learning objects. IMS CP is mainly used as a packaging structure format for export, transfer and import of learning objects. In the prototypes, a content package containing the tools associated with a user’s workspace is downloaded to the browser when the user logs in, and one or more workspaces are initiated and started in the web browser. The reason that an RDF-binding was used was the need to represent more complex relations than is allowed by IMS CP and the use of RDF allows a workspace structure representation to be

extended⁴⁷ in order to represent complex relations and to take advantage of the machine-processable semantics of RDF.

One major benefit of using RDF as the native format for representing workspace structures and relations is that RDF makes it possible to dynamically connect workspace representations to the semantic annotations of VWE Services and Tools. This basically creates the same characteristics as Semantic Web Services (discussed in chapter 3.11.4), allowing semi-automatic discovery and matching of tools, services and content, based on the utilization of machine-processable metadata semantics. This also facilitates the matching of content and components in the VLE to other sources of data, such as pedagogical instructions, which was illustrated in paper 6. The same mechanism allows the RDF representation of a workspace to be dynamically connected to the pedagogical process of the representation – a relation that is discussed further below.

One problem associated with representing workspace structures using an RDF binding to IMS CP is that the IMS CP metamodel describes a packaging structure based on a recursive container model, suitable to represent using XML, which is also the standard way to represent IMS CP. IMS CP can be expressed using RDF, but the expressive power of RDF cannot be utilized because of the significant differences between the two approaches (see chapter 3.9.4). While powerful machine-processable semantics can be expressed using RDF, all the semantics of XML lies in its hierarchical tree structure, and the intended use of XML is to represent syntax – not semantics. Similar problems were encountered by Nilsson, Palmér, and Brase (2003), regarding the RDF binding to LOM. A consequence of this incompatibility is that every possible representation of a workspace can be represented using RDF, but not using IMS CP and this is a bottleneck created by the compromise that occurred in the striving to be standard compliant. An alternative could have been to use IMS CP in conjunction with IMS LD, or EML as it was called at the time. This would have added expressive power, but because the IMS LD also uses an XML binding for its standard representation, the problem would have remained, although to a lesser extent. However, the use of IMS LD is likely to have provided a better foundation for future development, because one strength of the IMS LD is that it is based on an abstract conceptual model that is not bound to XML, which means that it could be expressed using RDF instead. The use of Semantic Web technology is emphasized by Koper (2004) as a preferable path for future releases of IMS LD. Koper argues that the increasing demands for adaptation, individualization, reuse and the increasing focus on self-organized learning networks will make semantic web technology crucial for representing pedagogical design (referred to as *Educational Modeling* by Koper (2004)) and pedagogical processes in the future.

Besides being used for representing workspace structure, RDF was also used to represent metadata (or annotations) about resources at different levels of the VWE Learning Object Taxonomy. In general, the more modularized a system becomes, the more sophisticated metadata seem to be needed in order to describe artefacts and services for both human and system purposes. VWE primarily uses metadata for three things: for annotations used to register and locate resources, for describing properties for humans, and for describing properties for use by the system. Even though RDF was mainly chosen for its machine-readable and machine-processable semantic abilities, it is RDF's ability to represent very simple, as well as very complex, data models that make the system flexible. The ability to represent machine-processable data semantics enhances adaptability; more importantly, this ability creates the foundation for adaptivity, because metadata can be used as input for

⁴⁷ This does of course break the compliance with IMS CP.

automatically (or semi-automatically) modifying the VLE, based on input from users, the system or the outcome of pedagogical activities.

Taken altogether, this creates an interesting potential in terms of connecting the VLE (or rather the components of the composition that forms the VLE) and the digital learning content, to the pedagogical process. This is discussed in more detail in the next chapter.

5.3 Representing the pedagogical process

The research results presented in papers 5 and 6 address research question 3 by suggesting an approach for connecting components in the learning environment to the curriculum, which represents a vital piece of the pedagogical instructions.

5.3.1 The Annofolio infrastructure

The research described in paper 5 resulted in a prototype (the Annofolio) that can be used to annotate XML-documents with RDF metadata. As described in chapter 4.4, the Annofolio is built on top of Semantic Web technology, and utilizes the Annotea W3C recommendation (Koivunen, Swick, José, & Prud'Hommeaux, 2002) to “point in” RDF annotations using XPointers. The suggested Annofolio methodology mainly accomplishes two important things: first, it introduces graph-based markup methodology for annotation of XML resources; and second, it separates the metadata markup (annotations) from the resource that is subjected to markup. This approach is opposed to document centric XML-based approaches, like TEI (Ide & Sperberg-McQueen, 1995). Besides adding to the general problems associated with XML-based metadata (see, e.g., Nilsson, Palmér, & Brase, 2003; Nilsson, Naeve, Duval, Johnston, & Massart, 2008), approaches such as TEI require write-access to the annotated resource. This means that the access to the resource by definition becomes authoritarian, and that it is not suitable for contributing to a metadata ecology, such as the one suggested by Nilsson, Palmér, & Naeve (2002). Besides, even if everyone who wanted to annotate a resource could be given write-access, the resource would pretty soon be a complete mess of XML-tags and it would become unmanageable very quickly.

By handling semantic markup of XML resources as annotations that are separated from the resource, they can be managed as an integrated part of the metadata ecology. The use of RDF makes it possible to utilize the expressive and semantic capabilities of RDF as well as related Semantic Web technologies, such as ontologies.

Similar conclusions have also been made by others (see, e.g., Handschuh, Staab, & Ciravegna, 2002; Ver et al., 2003). Patriarche et al. (2005), who introduce a tool (NOESIS) that is based on principles that are similar to the Annofolio. In that respect NOESIS verifies the validity of the results presented in paper 5. There are some major differences though. The most important difference is that the Annofolio is implemented solely using web technology and is independent of any client – with the exception of the web browser, while NOESIS depends on client software. The Annofolio implements an Ajax-like approach, using ECMAScript (ECMA, 1999) for application logics on the client.

While NOESIS has a specific community focus, targeting health professionals and knowledge management in the medical field, the Annofolio is a generic tool that has no bindings to any specific application field or ontology. One of the objectives of the Annofolio is to support annotations that use both formal ontologies and non-formal annotations. This aim, together with the characteristic of being unbound by the resource that is annotated by storing annotations separated from resources, as well as separated from the Annofolio system in an external RDF compliant store, contributes to making the Annofolio suitable for contributing to a global and decentralized metadata ecology similar to notion discussed by Nilsson,

Palmér, & Naeve (2002). This was also discussed as a part of my theoretical framework (see chapter 3.9). A consequence of the methodology used by the Annofolio is that different contributors may independently add annotations, according to different ontologies, for different purposes, and in different contexts, to the metadata ecology without interfering with each other. This is referred to as different *metadata layers* in the software architecture, which is described in paper 5.

A differentiation between different types of metadata markup was also made in papers 5 and 6. The separation was made between internal/structural markup⁴⁸, external markup⁴⁹, and internal markup⁵⁰. A similar (but different) differentiation was made in the NOESIS project, where a separation was made between annotation types for cataloguing, conceptual annotations and textual annotations. While the Annofolio focuses on how and where metadata are managed, NOESIS focuses on the type and purpose of the annotations. This is likely to be a consequence of NOESIS focusing on a specific application area, while the objective of the Annofolio is to be neutral by only providing the infrastructure for annotations of XML resources.

XML-based approaches, like TEI, have as their field of application to *specify encoding methods for machine-readable texts, chiefly in the humanities, social sciences and linguistics* (Ide & Sperberg-McQueen, 1995), a field where most of the resources are texts that are relatively stable and where the markup is done by a few trusted individuals in a controlled environment, i.e., an authoritarian approach. However, such an approach does not scale well in a global and distributed environment where markup and annotations are added independently by a large number individuals, according to different ontologies, in different contexts and where annotations may overlap each other in terms of scope within the annotated resource. Besides being more or less impossible to manage using document centric approaches, the result would be a veritable mess. In addition, the mere nature of an infrastructure such as the Internet creates a need to go from machine-readable to machine-processable – in order to be human-manageable in the larger perspective. This is also the very essence of the Semantic Web as it was outlined by Berners-Lee, Hendler, & Lassila (2001) and Berners-Lee & Fischetti (1999).

One very interesting possibility opened up by the Annofolio is the ability to connect annotations of resources to social networks using the semantic capabilities of RDF. One example is by the use of the *Friend Of a Friend (FOAF)* ontology (Brickley & Miller, 2007). Such approaches have been described by Avram (2006), as well as by Macgregor & McCulloch (2006) and could very well be utilized together with the Annofolio, even though this has never been tested. The potential is there, and it is obviously interesting for TEL. In the context of this thesis, it would be a possible approach to dynamically connect pedagogical designs and pedagogical processes to components of the VLE and to digital learning resources as well as to the social context of learning – whatever that may be in whatever situation – using the semantics represented by RDF annotations and (potentially) associated ontologies. Koper (2004) refers to similar ideas in terms of *Self-Organized Learning Networks*. He argues that Semantic Web technology, together with representations of pedagogical design, such as IMS LD will play a central role for their accomplishment. One issue that is often raised, together with the reluctance to metadata tagging, is the lack of

⁴⁸ XML syntax containing structural markup according to the XML Schema

⁴⁹ Referring to markup addressing the resource as a whole, such as for cataloguing purposes or annotations referring to the complete resource.

⁵⁰ Referring to annotations, or RDF statements, addressing only a part of a resource with the purpose of replacing the semantics expressed using XML markup in the TEI example.

integrated tools for the Semantic Web. Even though there is still a long way to go, the Annofolio provides one piece of this puzzle.

Another problem that was discussed in paper 5 is the ability to manage resources that change. When an infrastructure that allows annotation of resources that are out of control of the annotator is put in place, there is no way of controlling whether resources change or not. This obviously needs to be dealt with somehow. Some minor tests were carried out in order to test the prototype's ability to cope with changes made in annotated XML-files. The results showed that changes were handled quite well, which are probably a result of the way XPointers were implemented. The XPointers' position is determined relative to the closest tag in the XML DOM tree. This means that annotations become immune to changes of the data – provided that this is done in the “right” section of the XML file, while annotations are most likely to be affected by data change in some cases. This needs to be evaluated further and possible problems need to be addressed in the future.

I will discuss how the Annofolio was put into practice in the following chapters. The research presented in paper 6 suggests how semantic annotations can be utilized within TEL and how it can be utilized to address research question 3 by suggesting a methodology for connecting pedagogical instruction to digital learning content.

5.3.2 Annotating the pedagogical process

The research in paper 6 examines how the Annofolio methodology can be used to represent pedagogical instruction, in this case represented by the Swedish national steering documents; of which the national curriculum and course plans are vital parts. The results that were presented in paper 6 are based on a rather limited experiment in curriculum mapping and more research needs to be carried out to verify the findings. In spite of that, the experiment gives a couple of indications of the potential of the Annofolio, and above all an indication of the potential of Semantic Web technology and the use of semantic annotations in TEL. The research presented in paper 6 is strengthened by recently published research (see, e.g., Dietze, Gugliotta, & Domingue, 2007; Koper, 2004; Psyche, Bordeau, Nkambou, & Mizoguchi, 2005; Devedzic, 2004; Richter, Nejd, & Allert, 2002; Lama, Arroyo, Sicillia, & Lópes-Cobo, 2006), where the importance of the role of the Semantic Web technology for TEL is emphasized.

The ontology that was used for the experiment⁵¹ in paper 6 was very simple and was originally applied for the TEI encoding of the same documents. Most likely, this does not influence the results presented in paper 6, and it may even be an advantage in creating a controllable environment as the complexity is kept down and action and reaction can more easily be determined when using a simple ontology. The basic principles of how annotations of the curriculum can be used are still illustrated and details of the final ontology to be used for such a purpose are outside the scope of this thesis.

The underlying idea of the experiment was to apply a set of basic annotations to course plans. This basic set of annotation was comparable to a traditional authoritarian metadata approach, using a formal ontology, which could then provide “hooks” to which other metadata could be connected by semantic mapping of metadata from different sources.

In addition to these basic annotations, “anyone” can contribute to the metadata ecology in order to create an information infrastructure with the same intended multi-purpose use as the basic set of annotations – in this case for mapping the curriculum to digital learning resources based on central concepts in the curriculum.

⁵¹ The ontology (that was really a taxonomy) was described in detail in paper 6.

By implementing a simple prototype that utilized the Annofolio annotation infrastructure and the provided RDF annotations, it became possible for teachers and students to locate suitable digital learning resources based on annotations in the curriculum. One effect of this was that it also became possible to locate resources intended for a completely different subject. See example in chapter 4.5. This effect is a result of the dynamic nature of machine-processable semantics and its ability to help in (and even automate) locating and matching needs with resources. In fact, it has a much greater potential in a wider perspective than was illustrated by this rather rudimentary experiment, and this potential is, for example, discussed by Koper (2004), and by Mohan & Brooks (2003), and even more explicitly by Lama, Arroyo, Sicillia, & López-Cobo (2006), where an ontology that connects learning objects to Semantic Web Services is suggested based on the Web Service Modeling Ontology (WSMO) and the Web Service Modeling Language (WSML) (Roman et al., 2005). This is the approach taken by the LUISA project⁵², where competency gaps are matched with suitable learning objects with the help of WSMO ontologies described using WSML. The research presented by Lama et al. (2006) is especially relevant as it contributes to a formalized approach to connecting Semantic Web Services to learning Objects – something that is still missing in the VWE Architecture Framework, where a much simpler approach was used that does not allow the same level of automatization. The main reason for this is that the notion of Semantic Web Services did not occur until very late in the project and the focus was on other modularity issues, which were discussed in previous chapters. In addition, the field of Semantic Web Services is still a young and rather immature field and it is not yet possible to clearly determine in which direction things are moving, but it seems pretty clear that the experiences made in the field will have a great impact on and be beneficial for TEL. The potential is especially appealing in the light of Semantic Web based representations of pedagogical design and pedagogical processes in relation to modular VLEs and modular digital content in a global, distributed environment where a manifold of pedagogical approaches need to coexist (see, e.g., Naeve & Sicilia, 2006; Monceaux et al., 2007). I will return to this discussion in chapter 5.4 where I attempt to pull things together.

One of the strengths of the methodology, used by the Annofolio, is that it allows a combination of formal ontologies, such as the simple example given in paper 6, and less formal social annotations that may evolve in the metadata ecology, much like the idea of *Folksonomies* (Voss, 2007). Even though formal ontologies can be usable, they are only used to a certain extent and there is a need to allow less formal approaches as a complement. One example of how such approaches could be utilized is to allow teachers to annotate their use of digital learning resources. Teachers could also be allowed to make annotations to the curriculum, which could then be used by other teachers in other contexts. Teachers' annotations could also be used as a way of connecting different parts of the pedagogical design to various components of TEL.

The Swedish school system, which is extremely decentralized, has a continuum of levels in the curriculum. The documents that represent the curriculum range from the national steering documents, containing the national curriculum, national course plans etc., via local steering documents at the municipal level (containing more detailed and locally adapted course plans and other curriculum documents), to documents at the individual school level, containing detailed course plans for implementation, and equivalents to instructional designs in the shape of lesson plans and likewise. The lesson plan as a metaphor for instructional design was discussed in chapter 3.4. While the latter documents, which represent the pedagogical designs that are supposed to be implemented through teachers' practice, may be represented using

⁵² <http://www.luisa-project.eu>

approaches such as IMS LD, all the different levels of the curriculum need to be semantically interconnected through an infrastructure, such as the one made possible by the Annofolio.

Furthermore, there is a need to interconnect this infrastructure with a standard, efforts such as IMS LD, in order to make it an integrated part of the system representation of pedagogical design – or learning design, as Koper (2005) refers to it in the context of IMS LD. One possible way to approach this issue would be to allow Semantic Web annotations of IMS LD. This would work in a similar fashion as the annotation of course plans presented in paper 6. Even though the preferable way forward would be to base IMS LD itself – or future equivalents of IMS LD – on Semantic Web technology, which has also been suggested by Koper (2004). This would help in solving many of the issues encountered in this thesis.

There are many factors that indicate that Semantic Web technology will have a significant impact on TEL in the future. In the context of this thesis, Semantic Web technology can be regarded as the (dynamic) glue that keeps everything together in a semantically interconnected and dynamic web of components.

The Semantic Web approach to curriculum mapping, suggested in papers 5 and 6, is only the beginning of the exploration of this field and a lot of research still needs to be done. Research question 3 has been addressed, but only partially answered. The issue of representing pedagogical designs and pedagogical processes opens up an abyss with endless potential opportunities, and a huge number of occurring problems, some of which were discussed above, related to other research in this area. However, the results presented in papers 5 and 6 provide indications of how to proceed and of future needs that need to be regarded from a wider perspective. This wider perspective will be briefly discussed in the next chapter, where I will also summarize my research results and discuss them in relation to each other.

5.4 Final discussion and conclusions

This thesis spans four aspects of TEL that are of interest in the light of my three research questions. These aspects were outlined and illuminated in my theoretical framework as being about: *pedagogy*, *technology*, *content*, and *learning activities*. Each one of these aspects contains their respective components that were discussed in chapter 3 and to various extents addressed by my research questions.

5.4.1 The context of my research contributions

The essence of the research presented in this thesis concerns the interdependence between learning theories, the implementation and representation of pedagogical design (traditionally instructional design), pedagogical process, the design and implementation of learning technology and digital learning content. These are all interconnected in a complex network of mutual dependencies, and changes in one place will most likely trigger changes and/or trade-offs in other parts of this complex network.

The mapping between the desired pedagogical approach and the corresponding learning activities is often weak and insufficient when using technology as a mediator. This mismatch is often caused by trade-offs that are constantly made when designing and implementing learning technology.

The distributed nature of the web and a modular learning infrastructure approach, together with an increasing focus on the collaborative and social aspects of learning, emphasize the need for a better mapping between learning technology and learning theories (see, e.g., Schluep, Bettino, & Guttormsen Schär, 2005; Olivier, Roberts, & Blinco, 2005; Counterman et al., 2005). In addition, the introduction of social and collaborative software makes the

pedagogical process even more unpredictable, which in turn makes it harder to design and represent pedagogical design and process.

So far, the most common strategy to deal with VLEs for Web-Based Learning has been to implement general-purpose, web-based learning platforms, i.e., LMSes. One of the paradoxes of LMSes is that the concept was first invented to be used together with learning objects, and the paradox lies in the fact that LMSes provide a non-modular learning environment for modular learning content, an issue that was discussed in chapter 5.2 and 3.7.1.

The idea of general-purpose LMSes appears to be appealing in many ways, but a problem is that LMSes stand out as being too isolated and restricted in the wider perspective – it is like putting the learners behind a fence so that they should not be tempted by what resides outside. LMSes tend to be implemented in a silo-like fashion that prevents interaction with the outside world and implementations are often based on old-fashioned conceptions of the requirements of TEL. These old-fashioned conceptions emanate from previous practice and legacy of concepts like Computer-Based Training (translated into Web-Based Training on the web), and similar approaches that are often a non-fit to the requirements of a global world. Today, education and learning are becoming a global affair with a strong focus on learner-centered and personalized learning – very much the opposite of what is supported by many LMSes.

By necessity, the learning environment becomes much more complex when the step is taken from the local, controllable environment of Computer-Based Training to the global, distributed and much less controllable learning environment on the web. The learning environment is extended from covering only what is provided by the teacher, in terms of software and content, to encompassing whatever is available in terms of content and services on the Internet.

Teachers can no longer solely decide what students should be given access to and there is always the possibility of choosing alternative ways to knowledge through the use of alternative content, services and through interaction with others. This situation complicates the every day life of teachers, but at the same time it is an unequalled pedagogical opportunity and a reality that needs to be dealt with in various ways. The situation does not only complicate the life of teachers, it complicates the life of learners as well. Suddenly everything is at the tip of their fingers – endless possibilities in terms of information, learning content, social interaction, services and so on, and this situation increases the need for durable learning strategies and durable learning tools that are loosely coupled. Ultimately it will be a challenge for teachers to adapt their teaching strategies so that they conform to this new situation, and likewise it will be a challenge for learners to adapt their learning strategies accordingly. This is also a challenge for learning technology to enabling teachers and learners to keep it all together in order to create manageable learning situations rather than a confusing chaos.

The effect of globalization is reinforced by the increasing focus on phenomena like personalization, flexible learning, life-long learning, on-demand learning, distance learning, and the disappearance of educational system borders (such as between stages and levels), and other trends in TEL (see, e.g., Howell, Williams, & Lindsay, 2003). At the same time, the pace of the technological development is so fast that there is no way that LMS vendors can keep up and always be in the frontline, implementing all the new functionality that is needed for all desirable learning scenarios. Instead, LMSes become like Swiss army knives – they are quite good at doing a lot of things quite poorly. As a comparison: it is relatively unusual to see professional craftsmen who use Swiss army knives. They use a modular system with special tools for special tasks, and tools are combined in different ways in different contexts, addressing different tasks. Without a modular *learning infrastructure*, based on standards for interoperability, it will more or less be left to the LMS vendors to decide which tools will be

available, and by that they also decide which pedagogical approaches that will be supported. With a modular learning infrastructure on the other hand, special-purpose software can be composed in a loosely coupled fashion to form VLEs that are much more powerful and at the same time flexible. This was illustrated by the research presented in papers 2, 3, and 4.

The situation described above is not only the reality for TEL, it is also the reality within many other areas where ICT is used, and it is also the main reason for the general ongoing development toward modular infrastructures, where new and better services (added value) are created by combining services from different service providers – the web is turning into a gigantic “building kit” that is sometimes described using atoms, Lego, puzzles or any other of the many popular metaphors.

While this development creates a number of interesting new possibilities, it also introduces a number of new issues and problems that need to be solved. One of the most urgent problems concerns how to keep everything together in the highly distributed and loosely connected infrastructure that becomes the result. There is a need for sophisticated mechanisms to “keep things together” in a way that makes the VLE appear as a coherent whole in an environment where the “whole” is actually more than the sum of its parts – it is the sum of its part, plus their mutual relations – and where different services, and data from different sources can be seamlessly stitched together, using modular approaches and Semantic Web technology. This is also the context within which this thesis contributes to TEL as a research area.

5.4.2 An eclectic summary

The theoretical framework (see chapter 3), together with a weighted analysis of the results presented in the appended papers, can be used to pinpoint a chain of interrelationships that lead to a couple of vital conclusions about the research results presented in this thesis, but also about things that need to be further explored and discussed.

First of all, the part of my theoretical framework that originates from the pedagogical field clearly shows that there are different theories about how we learn, and that these theories are applicable to different pedagogical contexts, often motivated by different learning objectives and types of skills or knowledge to be acquired. In addition, the pedagogical methods chosen by teachers reflect their views on learning, and by that (consciously or not) the learning theories they believe in. The pedagogical methods favored by teachers affect the way in which pedagogical instructions and processes are expressed. This has consequences for how pedagogical design (instructional design) and process need to be applied. In chapter 3.4, I discussed how different learning theories put different demands on the practice of instructional design. Consequently, this affects the ways in which instructional design must be represented in VLEs. This is directly related to the view of the learning process in each theory. System representations that only support sequences of activities, such as those used with a chunking and sequencing approach, may be satisfying in a pedagogical situation where a behavioristic approach is used. Behaviorists regard the pedagogical process as a sequence of predictable activities, leading to a certain outcome in terms of changes in learners’ behavior – changes that can then be tested in steps in the sequence. Such processes are rather straightforward to represent using a hierarchal structure, suitable for representation using XML.

The development within TEL has led to a change in technology focus, in terms software and service, as well as to a change in pedagogical focus. With technologies that altogether better support collaboration, social interaction, and a sophisticated organization of learning content, it also becomes natural to emphasize pedagogical approaches that have the potential to make better use of those developments. Social software and collaborative learning environments

create the means for pedagogical approaches and learning that better support the type of learning that requires a higher level of cognitive processing targeting learning objectives and contexts where many teachers favor constructivist approaches. This is also likely to be one of the main reasons that many of the discussions about TEL during the last ten years have addressed issues on how to better match TEL to the requirements of constructivism. This is especially apparent regarding matters that concern the relations between instructional design, constructivism and TEL (see, e.g., Mayer, 1999; Schank, Berman, & Macpherson, 1999; Jonassen, 1999). Those issues were also one of the major driving forces behind the emergence of EMLs, such as the IMS LD, as well as the ongoing change of focus from instructional design to learning design – concepts that I refer to by using the general and collative term: pedagogical design.

The necessity of the above discussion is clearly manifested by the fact that the situation becomes much more complicated in pedagogical scenarios where constructivist approaches are used. While the learning process can quite easily be predicted in a behaviorist scenario, it becomes virtually unpredictable in a constructivist scenario. The unpredictability expresses itself in different ways, and depending on the choice of methodological approach, it influences different parts of the learning environment and the learning situation in different ways. It may, for example, be hard to predict exactly what content students will need or use, it may be hard to predict what tools and what kind of VLE students will need, and it will certainly be hard to predict exactly what learning path(s) students will take to acquire their desired knowledge. These are all things that were kept together by the instructional design and its system representation in the behaviorist scenarios. However, that requires education and learning to be predictable and curriculum-centric to an extent that it is not possible in a typical constructivist scenario. Hence, the pedagogical design and processes need to be represented in a much more dynamic and self-organizable way.

Therefore, the VLE must be adaptable in order to enable the utilization of such dynamic and self-organizable representations of the pedagogical design and process. Experiences from computer science, as well as from TEL show, simultaneously, that modularity promotes flexibility as modules in a modular system can be composed (and decomposed) to form compositions that have different properties and qualities for use in different contexts. This was discussed in chapters 3.6 and 3.11 of my theoretical framework. One conclusion that was drawn was that an important step toward the needed flexibility could be made by the utilization of modular approaches for TEL. This is evident from the success of object-orientation and modularity within computer science and software engineering. In addition, there is already a widespread and widely accepted model for modular digital learning content, i.e., learning objects. A feasible approach would therefore be to extend the learning object model to also embrace the VLE. As a matter of course, this hypothesis lead to the formulation of research question 1 and 3.

The findings presented in paper 1 showed that there were some major technical issues that needed to be addressed, which constituted that basis for the research in paper 2, where the VWE Learning Object Taxonomy was formulated. This taxonomy demonstrated that it was theoretically possible to place digital learning content and the VLE in a common conceptual model. The research presented in papers 3 and 4 clearly demonstrated that the taxonomy could be translated into a modular and distributed software architecture.

Together those research results provide the basis for a very flexible and modular infrastructure that embraces both the VLE and digital learning content. This infrastructure was implemented by the VWE prototypes. The flexibility lies foremost in a very high level of adaptability. Both the VLE and the digital learning content is fully modular and can be composed and configured to suit most requirements – provided that the needed tools exist, or

can be developed. The underlying technology and infrastructure provide the conditions needed to participate in a global infrastructure, utilizing components and services provided by others. Similar approaches are already being used related to the Web 2.0 phenomenon, where lightweight technologies such as RSS and XML-RPC (and sometimes proprietary APIs) are used to accomplish mash-ups, which can be considered to be equivalents of simple composite applications.

The possibilities created by this kind of composite VLEs will most likely change the way in which VLEs are regarded. Currently the VLE is often viewed as something that exists in terms of a system with well-defined system borders, often in terms of an LMS product. The antithesis to the LMS occurs when a modular approach is applied to its extreme. In such cases, the VLE does not actually exist until it is created, and it will always be created for a specific purpose, aiming to address the requirements of a specific learning situation, associated with specific pedagogical and functional needs. The only thing that is pre-existing is the modular infrastructure for composing VLEs and content – *the learning infrastructure*.

However, even though a highly adaptable learning infrastructure creates the possibility to compose a VLE “on demand”; a VLE that can be continuously adapted and changed, there is still a major issue that needs to be addressed and that is adaptivity. Even though the VLE is adaptable and can be adapted at run-time, this still assumes a certain amount of predictability, in terms of needed tools, learning content and structure.

There is an important difference between the notion of adaptability and adaptivity. An adaptable VLE can be adapted by actors in the system, and in this case it means that functionality and content can be added and removed from the VLE at any time from composition-time and forward. This involves conscious actions by a user, or something that has been pre-determined by a pedagogical designer. Adaptivity on the other hand occurs when the VLE and the content has the ability to change automatically based on changes or feedback from users in the system, triggered by processes and activities – such as changes in the pedagogical process, that in turn may be triggered by the choice of a user, user profiles (such as previous experience) or other kinds of feedback.

There is a need to connect to other components within the learning infrastructure, based on the semantics expressed and matched to different components in the learning infrastructure. Such connections were illustrated by the experiments presented in papers 5 and 6, where Semantic Web technology was suggested as the approach to accomplish such dynamics. In order to make the VLE and the digital learning content adaptive, the pedagogical design must be represented in a way that facilitates adaptation to changes in the process based on feedback from users, from the VLE, from content, or even from the pedagogical process itself. One approach to this is the notion of *attention metadata* (see, e.g., Najjar, Wolpers, & Duval, 2006). Attention metadata can be used to represent the relationship between users and other resources within TEL. Attention metadata are harvested from different sources and is compiled into extensive data about users actions, activities, preferences etc., which means that it is suitable as input to the pedagogical process.

Even though the experiments that were presented in papers 5 and 6 were quite small, and restricted to the context of adding annotations to the curriculum, they indicate that Semantic Web technology and annotations in a metadata ecology can be taken much further. There is some recent research that supports this thesis, such as, by Nilsson et al. (2002), who emphasize the importance of a metadata ecology for learning, by Koper (2004), who discusses the use of semantic web technology for representing IMS LD, and by Psyche, Bordeau, Nkambou, & Mizoguchi (2005), who discuss an ontology approach to connect

learning theories to learning design (see also, e.g., Nilsson et al. 2008; Nilsson, Johnston, Naeve, & Powell, 2007; Powell, Nilsson, Naeve, Johnston, & Baker, 2004; DCMI, 2007).

By adding the ability to represent pedagogical process and pedagogical instruction that utilize machine-processable data-semantics, it also becomes possible for different components in the learning infrastructure to interpret the semantics and act upon it. By using the same approach for other parts of the learning infrastructure, such as digital learning content, Semantic Web Services, Semantic Web annotations etc, it becomes possible to connect and integrate data from different sources – using technologies such as machine reasoning – in order to respond to actions and trigger changes that can help in creating reasonably self-organizable learning environments.

The above discussion indicates that while modularity creates the pre-conditions for adaptability, Semantic Web technology creates the accompanying pre-conditions for adaptivity. It should be emphasized that the idea is not that the system should create new pedagogical design, but that the system, based on the input-types described above, can match learning situations to reusable learning designs, for example, based on patterns for re-occurring learning scenarios.

5.5 A complex path for the future

A re-occurring issue, related to the research presented in this thesis, is complexity. Complexity encompasses complex learning and complex cognitive processing. Complexity in terms of learning technology and infrastructure is also included, in combination with globalization and a constantly increasing focus on interdisciplinary approaches, as well as a changing view of learning and knowledge. Finally, there is complexity in terms of complex relations between different components of TEL. This was emphasized in my theoretical framework and illustrated by Figure 3-13.

Several concepts and ideas that have been discussed in this thesis, such as modularity loose coupling, self-organizability, adaptivity, dynamics, are concepts explored within Complexity Theory:

“Complexity science focuses on the dynamic relationships and patterns among phenomena rather than the static properties of isolated objects. Its explanatory framework and core concepts—which include emergence, evolution and embodiment—have been adopted by mathematicians, biologists, physical scientists and social scientists, as well as by many thinkers in the humanities.”⁵³

Consequently, understanding complexity seems to hold the key to understanding the embodiment of TEL, and to understand how to utilize modular learning infrastructures to achieve adaptable as well as adaptive systems for the future. My conclusions in this thesis include the views that modularity creates adaptable systems, and that the use of Semantic Web technology may help in creating the flexibility needed for adaptivity – a point of view that is also verified in other studies (discussed in chapter 3.9 and 5.4.2).

However, the interrelations between the different components (technical, social, pedagogical and so on) are still unclear in many respects and a lot more research needs to be done in order to take the next steps. In order to realize the learning environments of the future and to develop better and more suitable learning technology, there is clearly a need to better understand the interrelations of the different components of TEL, and the emergence of the huge and complex system that learning and education become a part of. It will not be possible

⁵³ Cited from the *Complexity and Education* web site at
http://www.complexityandeducation.ualberta.ca/glossary/g_complexsci.htm

to continue to uphold the distinction between pedagogy and technology that is made far too often. Pedagogy and technology are both parts of the same complex system, which has been illustrated in this thesis. My hypothesis is that all aspects of learning must be regarded and analyzed as one modular and nested system, where everything is linked to everything else, and where all parts may influence all of the other parts – either explicitly or implicitly.

A modular view of TEL, such as the view put forward in this thesis, does not mean that an approach based on reductionism would be appropriate. Such a view would be far too unsophisticated, and would most likely disregard some of the most important aspects of TEL. Considering this point of view, Complexity Theory is suggested as a suitable point of departure. Even though it is a fairly new scientific approach, it has already been applied to several research areas, such as social science, biology, education (see, e.g., Brent & Simmt, 2003), and Knowledge Management, (see, e.g., Dann & Barclay, 2006), and it stands out as being suitable for multidisciplinary and eclectic research approaches – whose importance was emphasized in the introduction of this thesis.

Complexity Theory may be used as an approach to understand and explain many of the issues at a *macro-level*, but there is still a need to understand relations and issues that may occur at the *micro-level*. While the macro-level represents the complex system that is TEL, with all its dependencies, the micro-level represents the environment and the components that are used in a specific context, such as a learning situation (and related activities), that is interrelated to the complex system at the macro-level. The relations and interactions at the micro-level are less complex in that they are more clearly demarcated, and by that more manageable, which makes it possible to regard them as an activity system. The micro-level is also more predictable to some extent. However, in the same sense that the VLE does not exist in a modular environment until someone creates it, the context of the micro-level does not exist until it is created. This is a consequence of regarding the micro-level in the context of being a part of a complex system. This rather complicated reasoning indicates that Complexity Theory alone cannot solve all problems, and that other theories may be more suitable in some cases, and especially at the micro-level. One such theory is Activity Theory (Engeström, 1999), which is (like constructivism) rooted in Vygotsky's sociocultural research.

Activities are modeled within the framework of *activity systems*. To better understand the actions and activities going on within an activity system, activity systems are commonly modeled as a triad, whereas an activity is undertaken by a human (being the *subject*). This activity is started for a purpose (that is the *object*), for example, to solve a problem, and the activities carried out are usually mediated by some kind of tool or similar (that is an *artifact*). Mediation can also be accomplished by other kinds of artifacts that are not explicitly regarded as tools in all contexts, such as culturally established artifacts (such as language). An activity system usually exists in a social context, constructed within the framework of a community (Engeström, 1999).

Combining Complexity Theory and Activity Theory may provide a theoretical framework, and the tools to facilitate the understanding of the properties needed in order to represent pedagogical processes and design the VLEs of the future. Besides supplementing each other, it might also be possible to use the two theoretical frameworks to verify the relevance of research results by analyzing results using a combination of both frameworks.

References

- Abdallah, R., El Hajj, A., Benzekri, A., & Moukarzel, I. (2002, 18-22 August). *On the Improvement of course interoperability in E-Learning Models*. Paper presented at the International Conference on Engineering in Education (ICEE 200).
- Abernethy, K., Treu, K., Piegari, G., & Reichgelt, H. (2005). *A Learning Object Repository in Support of Introductory IT Courses*. Paper presented at the Proceedings of the 6th conference on Information technology education, Newark, NJ, USA.
- ADL. (2001, June 22). The SCORM Overview. *SCORM Sharable Content Object Reference Model*. Retrieved 11 September, 2003, from <http://www.adlnet.org>
- ADL. (2004). *SCORM 2004 2nd Edition, Overview*. Alexandria, VA: ADL.
- Anderson, L. W., & Krathwohl, D. R. (2001). *A taxonomy for learning, teaching and assessing: A revision of Bloom's Taxonomy of educational objectives* (Complete edition). New York: Longman.
- Anderson, L. W., & Sosniak, L. A. (1994). *Bloom's taxonomy: a forty-year retrospective. Ninety-third yearbook of the National Society for the Study of Education*. Chicago: The National Society For The Study Of Ed.
- Anderson, N. (2006). Tim Berners-Lee on Web 2.0: "nobody even knows what it means". Retrieved 30 September, 2007, from <http://arstechnica.com/news.ars/post/20060901-7650.html>
- Anderson, P. (2007). *What is Web 2.0? Ideas, technologies and implications for education* (No. TSW0701): JISC.
- Andersson, C. (2000). *Kunskapssyn och lärande - i samhälle och arbetsliv* (1 ed.). Lund, Sweden: Studentlitteratur.
- Antoniou, G., & Van Harmelen, F. (2004). A semantic web primer. In *Cooperative information systems* (pp. 61-104). Cambridge, Mass.: MIT Press.
- Arsanjani, A., & Holley, K. (2006, September 18-22). *The Service Integration Maturity Model: Achieving Flexibility in the Transformation to SOA*. Paper presented at the IEEE International Conference on Services Computing (SCC'06), Chicago, USA.
- Avram, G. (2006). At the Crossroads of Knowledge Management and Social Software. *Electronic Journal of Knowledge Management* 4(1).
- Baldwin, C., & Clark, K. (2000). *Design Rules: The Power of Modularity* (Vol. 1). Cambridge, MA: The MIT Press.
- Bandura, A. (1977). *Social Learning Theory*: Englewood Cliffs, N.J. : Prentice Hall.
- Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Englewood Cliffs, NJ: Prentice Hall.
- Bandura, A., & Walters, R. H. (1963). *Social Learning and Personality Development*. New York Holt, Rinehart and Winston, cop.
- Bannan-Ritland, B., Dabbagh, N., & Murphy, K. (2002). Learning Object Systems as Constructivist Learning Environments: Related Assumptions, Theories, and Applications. In D. A. Wiley (Ed.), *The Instructional use of Learning Objects* (1 ed., pp. 61-97). Bloomington, IA: Agency for Instructional Technology, and Association for Educational Communications & Technology.

- Battle, S., Bernstein, A., Boley, H., Grosof, B., Gruningen, M., Hull, R., et al. (2005). *Semantic Web Services Framework (SWSF) Overview*: W3C.
- Beckett, D. (2004, 10 February 2004). RDF/XML Syntax Specification (Revised). Retrieved 17 October, 2007
- Berners-Lee, T. (1998, 1998-10-14). Why RDF model is different from the XML model. Retrieved January 15, 2004, from <http://www.w3c.org/DesignIssues/RDF-XML>
- Berners-Lee, T., & Fischetti, M. (1999). *Weaving the Web: The original design and ultimate destiny of the World Wide Web, by its inventor* (1 ed.): Collins.
- Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities [Electronic Version]. *Scientific American*. Retrieved 10 october 2006 from <http://www.sciam.com/article.cfm?id=the-semantic-web>.
- Bitzer, D., Lyman, E., & Easley, J. (1965). *The Uses of PLATO: a computer controlled teaching system* (No. R-268). Illinois: Coordinated Science Laboratory, University of Illinois.
- Bjerg, J. (2000). *Pedagogik, en grundbok*. Stockholm: Liber AB.
- Bloom, B., & Krathwohl, D. R. (1956). *Taxonomy of Educational Objectives: The Classification of Educational Goals, by a committee of college and university examiners. Handbook I: Cognitive Domain*. New York, NY: Longmans.
- Boutell, T. (1996). *CGI Programming in C and Perl*. New York: Addison Wesley.
- Bratina, T. A., Hayes, D., & Blumsack, S. L. (2002). Preparing Teachers To Use Learning Objects. *Technology Source* (Nov-Dec 2002).
- Bray, T., Paoli, J., & Sperberg-McQueen, C. M. (1998). Extensible Markup Language (XML) 1.0 [Electronic Version], 37. Retrieved 10 June 2007 from <http://www.w3.org/TR/1998/REC-xml-19980210.pdf>.
- Brent, D., & Simmt, E. (2003). Understanding learning systems: Mathematics education and complexity science. *Journal of Research in Mathematics Education*, 34(2), 137-167.
- Brickley, D., & Brickley, R. V. (2004, 10 February 2004). RDF Vocabulary Description Language 1.0: RDF Schema. Retrieved 10 October 2007, from <http://www.w3.org/TR/rdf-schema/>
- Brickley, D., & Miller, L. (2007, 2 November 2007). FOAF Vocabulary Specification 0.91. 0.91. Retrieved 18 November, 2007, from <http://xmlns.com/foaf/spec/>
- Bruner, J. (1961). The act of discovery. *Harvard Educational Review* (31), 21-32.
- Bush, V. (1945/1994). As We May Think [Electronic Version]. *The Atlantic Monthly*, July 1945. Retrieved 2007-03-01 from <http://www.press.umich.edu/jep/works/vbush/vbush.txt>.
- Capone, W., Christensen, C., Falls, J. M., Fletcher, D., & al., E. (2004). *SCORM: Sharable Content Object Reference Model - SCORM Sequencing and Navigation, Version 1.3.1*. Retrieved 2006-06-01, from <http://www.adlnet.org/>.
- Cardoso, J., & Sheth, A. P. (2006a). The Semantic Web and its applications. In J. Cardoso & A. P. Sheth (Eds.), *Semantic Web and Beyond: Computing for Human Experience Semantic Web Services, Processes and Applications* (pp. 3-33). New York: Springer Science+Business Media, LLC.

- Cardoso, J., & Sheth, A. P. (2006b). The Semantic Web and its applications. In J. Cardoso & A. P. Sheth (Eds.), *Semantic Web and Beyond: Computing for Human Experience* *Semantic Web Services, Processes and Applications* (pp. 3-33). New York: Springer Science+Business Media, LLC.
- CEdMA. (1995). CeDMA Sigs/Task Forces. Retrieved March 7, 2005, from http://www.cedma.net/sigs_task_forces.php
- Chou, C. (1999). Developing hypertext-based learning courseware for computer networks: the macro and micro stages. *Education, IEEE Transactions on*, 42(1), 39-44.
- Christiansen, J.-A., & Anderson, T. (2004). Feasibility of Course Development Based on Learning Objects: Research Analysis of Three Case Studies. *International Journal of Instructional Technology & Distance Learning*, 1(3).
- Clark, J. (1997, 15 December 1997). Comparison of SGML and XML. Retrieved 20 February, 2008, from <http://www-mit.w3.org/TR/NOTE-sgml-xml-971215>
- Clark, J., & DeRose, S. (1999). XML Path Language (XPath) Version 1.0, *W3C Recommendation 16 November 1999*: W3C.
- Clark, R. (2006). Building Expertise: Cognitive methods for Training and Performance Improvement (2nd ed.). UK: Jossey Bass Wiley.
- Collins, A. (1977). Processes in acquiring knowledge. In R. C. Anderson, R. J. Spiro & W. E. Montague (Eds.), *Schooling and the acquisition of knowledge* (pp. 339-363). Hillsdale, NJ: Erlbaum.
- Counterman, C., Golden, G., Gollub, R., Norton, M., Severance, C., & Speelmon, L. (2005, March 5, 2005). Sakai Java Framework. Version 1.0 RC5. Retrieved June 11, 2006, from <http://www.columbia.edu/acis/courseworks/sakai/externaldocs/Sakai-Java-Framework-05.pdf>
- Crowder, N. A. (1960). Automatic Tutoring by Intristic Programming. In A. A. Lumsdaine & R. Glaser (Eds.), *Teaching Machines and Programmed Learning: A source book* (1 ed., Vol. 1, pp. 286-298): National Education Association.
- Crowder, N. A. (1964). On the difference between linear and intrinsic programming. *Phi Delta Kappan*, (44), 250-254.
- Dann, Z., & Barclay, I. (2006). Complexity Theory and Knowledge Management Application. *The Electronic Journal of Knowledge Management*, 4(1), 11-20.
- DCMI. (2004, 2004-12-20). Dublin Core Metadata Element Set, Version 1.1: Reference Description. Retrieved 10 February, 2008, from <http://dublincore.org/documents/dces/>
- DCMI. (2007, 31 October 2007). Joint DCMI/IEEE LTSC Taskforce. Retrieved 10 February, 2008, from <http://dublincore.org/educationwiki/DCMIIEEELTSCTaskforce>
- Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., et al. (2000). The Semantic Web: the roles of XML and RDF. *Internet Computing, IEEE*, 4(5), 63-73.
- Dempsey, J. V., & Van Eck, R. N. (2007). Distributed Learning and the Field of Instructional Design. In R. A. Reiser & J. V. Dempsey (Eds.), *Trends and Issues in Instructional Design and Technology* (2 ed., pp. 288-300). Upper Saddle River, NJ: Pearson Education.
- Devedzic, V. (2004). Education and the Semantic Web. *International Journal of Artificial Intelligence in Education*, (14), 39-65.

- Dewey, J. (2004). *Individ, skola och samhälle : utbildningsfilosofiska texter*. Stockholm: Natur och Kultur.
- Dietze, S., Gugliotta, A., & Domingue, J. (2007). Context-Adaptive Learning Designs by Using Semantic Web Services. *Journal of Interactive Media in Education (Adaptation and IMS Learning Design. Special Issue)*, (September 2007).
- Dodani, M. (2002). The Dark Side of Object Learning: Learning Objects. *Journal of Object Technology*, 1(5), 37-42.
- Dodig-Crnkovic, G., & Crnkovic, I. (2003, January 2003). *Computing Curricula: Teaching Theory of Science to Computer Science Students*. Paper presented at the Hawaii International Conference on Education, Hawaii.
- Douglas, I. (2001, 10-13 October). *Instructional Design based on reusable learning objects: Applying lessons of Object-Oriented software engineering to learning systems design*. Paper presented at the ASEE/IEEE Frontiers in Education Conference, 31st annual, Reno, NV, USA.
- Downes, S. (2002, 11-13 November). *Design and Reusability of Learning Objects in an Academic Context: A New Economy of Education*. Paper presented at the eLearning: una sfida per l'universita, Milan, Italy.
- Dusek, V. (2006). *The Philosophy of Technology - An Introduction* (1 ed.): Blackwell Publishing Ltd.
- Duval, E. (2004). Learning Technology Standardization: Making Sense of it All. *Computer Science and Information Systems*, 1(1), 33-43.
- ECMA. (1999). Standard ECMA-262 ECMAScript Language Specification (Standard specification): ECMA.
- Edwards, D., Rai, S., Philips, R., & Fung, L. C. C. (2007). A Framework for Interoperable Learning Objects for E-Learning. In A. Koohang & K. Harman (Eds.), *Learning Objects and Instructional Design* (pp. 437-469). Santa Rosa, CA: Informing Science Press.
- Egidius, H. (2003). *Lärande för 2000-talet* (4 ed.). Stockholm: Natur och kultur.
- Ekengren, D. (1998). *Metadata on the Web using RDF*. Kista: SITI.
- Engeström, Y. (1999). Activity theory and individual and social transformation. In Y. Engeström, R. Miettinen & R.-L. Punamäki (Eds.), *Perspectives on Activity Theory* (pp. 19-38). Cambridge: Cambridge University Press.
- English, H. B. (1942). How Psychology can Facilitate Military Training: A Concrete Example. *Journal of Applied Psychology*, 3-7.
- Erl, T. (2007). *SOA Principles of Service Design* (1 ed.). Boston, MA: Prentice Hall.
- Ertmer, P. A., & Newby, T. J. (1993). Behaviorism, Cognitivism, Constructivism: Comparing Critical Features from a Design Perspective. *Performance Improvement Quarterly*, 6(4), 50-72.
- Fensel, D., & Bussler, C. (2002). The Web Service Modeling Framework WSMF. *Electronic Commerce: Research and Applications*, 1(2), 118-130.
- Fensel, D., Van Harmelen, F., & Davies, J. (2003). *Towards the semantic web ontology-driven knowledge management*. Chichester, England ; Hoboken, NJ: J. Wiley.
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. University of California, Irvine.

- Fisher, D. (2007, 2007). Instructional Design -- The Taxonomy Table. Retrieved February 10, 2008, from <http://oregonstate.edu/instruct/coursedev/models/id/taxonomy/>
- Fisler, J., & Bleisch, S. (2006, April 11-13). *ELML, The e-lesson markup language: Developing Sustainable e-Learning Content Using an Open Source XML Framework*. Paper presented at the WEBIST 2006 - International Conference on Web Information Systems and Technologies, Setubal, Portugal.
- Fletcher-Flinn, C. M., & Gravatt, B. (1995). The Efficacy of Computer Assisted Instruction (CAI): A Meta-Analysis. *Journal of Educational Computing Research*, 12(3), 219-241.
- Forehand, M. (2005). Bloom's Taxonomy: Original and Revised. *Emerging perspectives on learning, teaching, and technology* Retrieved September 12, 2006, from <http://www.coe.uga.edu/epltt/bloom.htm>
- Friesen, N. (2004a, 30 august - 1 September). *Learning objects and standards: pedagogical neutrality and engagement*. Paper presented at the IEEE International Conference on Advanced Learning Technologies, Joensuu, Finland.
- Friesen, N. (2004b). Three Objections to Learning Objects and E-learning Standards. In R. McGreal (Ed.), *Online Education Using Learning Objects* (pp. 59-70). London: Routledge.
- Gagne, R. M. (1962). *Psychological Principles in System Development*. New York: Holt, Rinehart and Winston, cop.
- Gagné, R. M. (1965). *Teaching Machines and Programmed Learning, III*. Washington DC: National Education Association.
- Gagné, R. M. (1977). *The conditions of learning* (3. ed.). New York: Holt Rinehart & Winston.
- Garrett, J. J. (2005). Ajax: A New Approach to Web Applications [Electronic Version]. *Javalobby*. Retrieved 20 February 2008 from <http://www.javalobby.org/articles/ajax/>.
- Garshol, L. M. (2003). Living with topic maps and RDF. Retrieved 18 October, 2007, from <http://www.web-services.gov/tmrdf.pdf>
- Gibbons, A. S., & Nelson, J. (2002). The Nature and Origin of Instructional Objects. In D. A. Wiley (Ed.), *The Instructional Use of Learning Objects* (1 ed., pp. 25-59). Bloomington IA: Association for Educational Communications & technology, and Agency for Instructional Technology.
- Goldfarb, C. F. (1991). Standards-HyTime: a standard for structured hypermedia interchange. *IEEE Computer*, 24(8), 81-84.
- Goldfarb, C. F. (1996). The Roots of SGML -- A Personal Recollection. Retrieved March 26, 2007, from <http://www.sgmlsource.com/history/roots.htm>
- Greeno, J. G. (1989). A perspective on thinking. *American Psychologist*, 44(2), 134-141.
- Grimnes, G. A. G., Edwards, P., & Preece, A. (2004). Learning Meta-descriptions of the FOAF Network. In *The Semantic Web – ISWC 2004* (Vol. 3298/2004, pp. 152-165). Heidelberg: Springer Berlin.
- Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2), 199-220.
- Hall, W. (1994). Ending the tyranny of the button. *IEEE Multimedia*, 1(1), 60-68.

- Handschuh, S., Staab, S., & Ciravegna, F. (2002, 1-4 October 2002). *S-Cream, Semi-automatic creation of metadata*. Paper presented at the European Conference on Knowledge Acquisition and Management, Sigüenza, Spain.
- Havice, B., & Galloway, C. A. (1998). Integrating HyperStudio Multimedia with Elementary School Curriculum: A Third Grade Multimedia Activity. *Technology and Children*, 3(2), 3-7.
- Haviland, V. S., & McCall, M. J. (1999). Transformation through Technology: How HyperStudio Updated Middle School Research. *The English Journal*, 89(1), 63-68.
- Hayes, P. (2002). RDF Model Theory. Retrieved February 14, 2003, from <http://www.w3c.org/TR/rdf-mt/>
- Haynes, D. (2004). *Metadata for information management and retrieval*. London: Facet.
- Hazael-Mas, D. (2003). RDF in Real Life - Some examples of RDF applications. Retrieved 17 november, 2007, from <http://www.w3.org/2003/Talks/semtour-athens-rdfapp/>
- He, H. (2003, 2003-09-30). What is Service-Oriented Architecture? Retrieved November 01, 2005, from <http://webservices.xml.com/lpt/a/ws/2003/09/30/soa.html>
- Henning, M. (2006). The rise and fall of CORBA. *Component technologies*, 4(5), 28-34.
- Hodgins, H. (2000, April 05, 2002). Everything you ever wanted to know about learning standards but were afraid to ask. *Learnativity*. Retrieved 7 August, 2005, from <http://www.learnativity.com/standards.html>
- Hodgins, W. (2001, December 17, 2004). LALO. What's that? Some history behind current Learning Standards Initiatives and some updates! Retrieved January 4, 2005, from <http://www.learnativity.com/lalo.html>
- Homme, L. E. (1960). The Rationale of Teaching by Skinner's Machines. In A. A. Lumsdaine & R. Glaser (Eds.), *Teaching Machines and Programmed Learning: a source book* (Vol. 1, pp. 133-136). Washington, D. C.: National Education Association of the United States.
- Horrocks, I., Patel-Schneider, P. F., & van Harmelen, F. (2003). From SHIQ and RDF to OWL: the making of a Web Ontology Language. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 1(1), 7-26.
- Howell, S. L., Williams, P. B., & Lindsay, N. K. (2003). Thirty-two Trends Affecting Distance Education: An Informed Foundation for Strategic Planning. *Online Journal of Distance Learning Administration*, 6(3).
- Hylén, J. (2002). *E-lärande som utmaning*. Stockholm: Utbildningsdepartementet.
- Ide, N. M., & Sperberg-McQueen, C. M. (1995). The Text Encoding Initiative: Its History, Goals, and Future Development. *Computers and the humanities*, 29(1), 5-18.
- IEEE. (2002, June 12). Final 1484.12.1-2002 LOM Draft Standard. from http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf
- IMS. (2001, 28 September). IMS Learning Resource Meta-Data Best Practice and Implementation Guide. Version 1.2.1 Final Specification. Retrieved November 11, 2007, from <http://www.imsproject.org/>
- IMS. (2003a, 20 January 2003). IMS Learning Design Best Practice and Implementation Guide: Version 1.0 Final Specification. Retrieved February 10, 2008, from <http://www.imsglobal.org/learningdesign/index.html>

- IMS. (2003, 03 March 2003). IMS Simple Sequencing Best Practice and Implementation Guide. *Version 1.0 Final Specification* 1.0. Retrieved November 10, 2007, from http://www.imsglobal.org/simplesequencing/ssv1p0/imsss_bestv1p0.html
- IMS. (2004, 04 October 2004). IMS Content Packaging Best Practice and Implementation Guide. *Version 1.1.4 Final Specification*. Retrieved November 11, 2007, from http://www.imsglobal.org/content/packaging/cpv1p1p4/imscp_bestv1p1p4.html
- IMS. (2005a, July 21). IMS General Web Services Base Profile. *Version 1.0 Public Draft Specification*. Retrieved August 28, 2005, from <http://www.imsglobal.org/gws/index.html>
- IMS. (2005b, Tue, 11 Oct 2005). IMS Learner Information Package Specification. Retrieved October 12, 2005, from <http://www.imsglobal.org/profiles/index.html>
- Ismail, J. (2002). The design of an e-learning system: Beyond the hype. *The Internet and Higher Education*, 4(3-4), 329-336.
- Iyengar, A., MacNair, E., & Nguyen, T. (1997, 3-8 November). *An analysis of Web server performance*. Paper presented at the Global Telecommunications Conference (GLOBECOM apos), Phoenix, AZ, USA.
- Jank, W., & Meyer, H. (1997). Didaktikens centrala frågor. In M. Uljens (Ed.), *Didaktik* (pp. 47-74). Lund: Studentlitteratur.
- Januszewski, A., & Yeaman, A. (2001). *Educational Technology: The Development of a Concept* (1 ed.): LIBRARIES UNLIMITED INC.
- JISC. (2003). Managing the future with MLEs [Electronic Version]. Retrieved February 02, 2003 from http://www.jisc.ac.uk/publications/publications/pub_mlesg.aspx.
- Johnson, R. T. (1985). Effects of Cooperative, Competitive, and Individualistic Goal Structures on Computer-Assisted Instruction. *Journal of Educational Psychology*, 77(6), 668-677.
- Jonassen, D. H. (1991a). Evaluating constructivistic learning. *Educational Technology*, 31(9), 28-33.
- Jonassen, D. H. (1991b). Hypertext as instructional design. *Educational Technology Research and Development*, 39(1), 83-92.
- Jonassen, D. H. (1991c). Objectivism vs constructivism: do we need a new philosophical paradigm. *Educational Technology Research and Development*, 39(3), 5-14.
- Jonassen, D. H. (1999). Designing Constructivist Learning Environments. In C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models: A New Paradigm of Instructional Theory* (Vol. 2, pp. 215-239). Mahwah, NJ: LEA.
- Klein, M., & Bernstein, A. (2001). *Searching for Services on the Semantic Web Using Process Ontologies*. Paper presented at the International Semantic Web Working Symposium (SWWS), Stanford University, California, US.
- Klinger, E. (1999). Thought Flow: Properties and Mechanisms Underlying Shifts in Content. In J. L. Singer & P. Salovey (Eds.), *At Play in the Fields of Consciousness: Essays in Honor of Jerome L. Singer* (1 ed., pp. 29-50): Lawrence Erlbaum Associates.
- Koivunen, M.-R., Swick, R., José, K., & Prud'Hommeaux, E. (2002). Annotea: Metadata based annotation infrastructure. Honolulu: W3C, presented at WWW 2002 Developers Day.

- Koper, R. (2004). Use of the Semantic Web to Solve Some Basic Problems in Education: Increase Flexible, Distributed Lifelong Learning, Decrease Teacher's Workload. *Journal of Interactive Media in Education*(6), 1-23.
- Koper, R. (2005). An Introduction to Learning Design. In R. Koper & C. Tattersall (Eds.), *Learning Design; A Handbook on Modeling and Delivering Networked Education and Training* (1 ed., pp. 3-20). Berlin Heidelberg: Springer verlag.
- Koper, R., & Manderveld, J. (2004). Educational modeling language: modeling reusable, interoperable, rich and personalized units of learning. *British Journal of Educational Technology*, 5(35), 537-551.
- Koper, R., & Tattersall, C. (2005). *Learning design: A Handbook on Modeling and Delivering Networked Education and Training*. New York: Springer Berlin Heidelberg.
- Kraan, W., & Wilson, S. (2003, 2002-10-01). Dan Rehak: SCORM is not for everyone. Retrieved January 10, 2006, from <http://www.cetis.ac.uk/content/20021002000737>
- Lama, d. N., Arroyo, S., Sicillia, M.-A., & López-Cobo, J.-M. (2006). *Making Learning Objects Useful with Semantic Web Services*. Paper presented at the eChallenges 2006: Exploiting the Knowledge Economy: Issues, Applications, Case Studies, Barcelona, Spain.
- Li, D., Lina, Z., Finin, T., & Joshi, A. (2005, 03-06 January). *How the Semantic Web is Being Used: An Analysis of FOAF Documents*. Paper presented at the 38th Annual Hawaii International, Hawaii.
- Lysaught, J. P., & Williams, C. A. (1963). *A Guide to Programmed Instruction*. New York: Wiley.
- Macgregor, G., & McCulloch, E. (2006). Collaborative tagging as a knowledge organization and resource discovery tool. *Library Review*, 55(7), 291-300.
- Maffeis, S., & Schmidt, D. C. (1997). Constructing reliable distributed communication systems with CORBA. *Communications Magazine, IEEE*, 35(2), 56-60.
- Makola, D., Sim, Y. W., Wang, C., Gilbert, L., S, G., & Wills, G. (2006, 23-26 May). *A Service-Oriented Architecture for a Collaborative Orthopedic Research Environment*. Paper presented at the WWW2006, Edinburgh, Scotland.
- Marton, F., & Booth, S. (2000). *Om lärande* (1 ed.). Lund: Studentlitteratur AB.
- Masie, E. (2003, November 2003). Making Sense of Learning Specifications & Standards: A Decision Maker's Guide to their Adoption. 2nd. Retrieved January 15, 2005, from <http://www.masie.com/standards/>
- Mayer, R. E. (1999). Designing Instruction for Constructivist Learning. In C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models: A New Paradigm of Instructional Theory* (Vol. 2, pp. 141-159). Mahwah, NJ: LEA.
- McCormick, R. (2003, August 26-30). *Keeping the Pedagogy out of Learning Objects*. Paper presented at the EARLI 2003.
- McGreal, R. (2004). Learning Objects: A practical Definition. *International Journal of Instructional Technology and Distance Learning*, 1(9), 21-32.
- McGuinness, D. L., & van Harmelen, F. (2004, 10 February). OWL Web Ontology Language Overview. Retrieved October 10, 2007, from <http://www.w3.org/TR/owl-features/>
- Meece, J. L. (2002). *Child and adolescent development for education* (2 ed.). New York: McGraw-Hill.

- Merrill, D. M. (1983). Component Display Theory. In C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models: an overview of their current status* (pp. 279-334). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Merrill, D. M. (1985). Where is the authoring in authoring systems. *Journal of Computer-Based Instruction*, 14(1), 7-11.
- Merrill, D. M. (1999). Instructional Transaction Theory (ITT): Instructional Design Based on Knowledge Objects. In C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models A new Paradigm of Instructional Theory* (1 ed., Vol. 2, pp. 397-424). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
- Merrill, D. M. (2002). First Principles of Instruction. *Educational Technology Research and Development*, 50(3), 43-59.
- Merrill, D. M., Li, Z., & Jones, M. K. (1991). Instructional Transaction Theory. *Educational Technology*, 31(6), 7-12.
- Meyer, S. R. (1960). Report on the Initial Test of a Junior High-School Vocabulary Program. In A. A. Lumsdaine & R. Glaser (Eds.), *Teaching Machines and Programmed Learning: a source book* (Vol. 1, pp. 229-246). Washington, D. C.: National Education Association of the United States.
- Mohan, P., & Brooks, C. (2003). *Learning objects on the semantic Web*. Paper presented at the The 3rd IEEE International Conference on Advanced Learning Technologies.
- Monceaux, A., Naeve, A., Sicilia, M.-A., Garcia-Barriocanal, E., Arroyo, S., & Guss, J. (2007). Targeting Learning Resources in Competency-based Organizations: A Semantic Web-based Approach. In J. Cardoso, M. Hepp & M. D. Lytras (Eds.), *The Semantic Web; Real World Applications from Industry* (pp. 143-167). New York: Springer Verlag.
- Monty, M. L., & Moran, T. P. (1986). A LONGITUDINAL STUDY OF AUTHORING USING NOTECARDS. *ACM SIGCHI Bulletin*, 18(2), 59-60.
- Mylov, P. (2000). Lärande i och om praktiken. In J. Bjerg (Ed.), *Pedagogik, en grundbok* (pp. 127-172). Stockholm: Liber AB.
- Naeve, A. (2001, October 24-26). *The Concept Browser - a new form of Knowledge Management Tool*. Paper presented at the 2nd European Web-based Learning Environments Conference (WBLE 2001), Lund, Sweden.
- Naeve, A. (2005). The Human Semantic Web – Shifting from Knowledge Push to Knowledge Pull. *International Journal of Semantic Web and Information Systems (IJSWIS)*, 1(3), 1-30.
- Naeve, A., & Sicilia, M.-A. (2006, 20-23 June). *Learning processes and processing learning: From Organizational Needs to Learning Designs*. Paper presented at the Adaptive Hypermedia conference (AH'06) Dublin.
- Nagarajan, M. (2006a). Semantic Annotations in Web Services. In J. Cardoso & A. P. Sheth (Eds.), *Semantic Web Services, Processes and Applications* (pp. 35-61). New York: Springer Science+Business Media, LLC.
- Nagarajan, M. (2006b). The Semantic Web and its applications. In J. Cardoso & A. P. Sheth (Eds.), *Semantic Web Services, Processes and Applications* (pp. 3-33). New York: Springer Science+Business Media, LLC.
- Najjar, J., Wolpers, M., & Duval, E. (2006, 23-26 May). *Attention Metadata: Collection and Management*. Paper presented at the 15th International World Wide web Conference, Edinburgh, Scotland.

- Nelson, T. H. (1965, August 24). *A file structure for the complex, the changing and the indeterminate*. Paper presented at the ACM Annual Conference/Annual Meeting, Proceedings of the 1965 20th national conference, Cleveland, Ohio, United States.
- Niederhauser, D. S., Reynolds, R. E., Salmen, D. J., & Skolmoski, P. (2000). The Influence of Cognitive Load on Learning from Hypertext. *Journal of Educational Computing Research*, 23(3), 237-255.
- Nielsen, J. (1995). Hypertext on the Internet. In *Multimedia and Hypertext: The Internet and Beyond* (1 ed., pp. 165-216). London: Morgan Kaufmann Publishers.
- Nierstrasz, O., & Dami, L. (1995). Component-Oriented Software Technology. In O. Nierstrasz & D. Tsichritzis (Eds.), *Object-Oriented Software Composition* (1 ed., pp. 3-28). Hertfordshire: Prentice Hall International (UK) Ltd.
- Nilsson, B. (2001). Socialkonstruktivism - positioner, problem och perspektiv (1 ed.). Malmö: Liber AB.
- Nilsson, M. (2001). The Semantic Web: How RDF will change learning technology standards [Electronic Version]. Retrieved April 11, 2006 from <http://zope.cetis.ac.uk/content/20010927172953>. Centre for Educational Technology Interoperability Standards (CETIS).
- Nilsson, M., Johnston, P., Naeve, A., & Powell, A. (2007). The Future of Learning Object Metadata Interoperability. In K. Harman & A. Koochang (Eds.), *Learning Objects: Standards, Metadata, Repositories, and LCMS* (pp. 255-313): Informing Science press.
- Nilsson, M., Naeve, A., Duval, E., Johnston, P., & Massart, D. (2008). ProLEARN Deliverable D4.7: Harmonization of Metadata Standards. *ProLEARN Deliverables*. Retrieved 10 February, 2008, from <http://ariadne.cs.kuleuven.be/lomi/images/5/52/D4.7-prolearn.pdf>
- Nilsson, M., Palmér, M., & Brase, J. (2003). The LOM RDF Binding - Principles and Implementation. *Proceedings of the Third Annual ARIADNE conference*.
- Nilsson, M., Palmér, M., & Naeve, A. (2002). *Semantic Web Meta-data for e-Learning - Some Architectural Guidelines*. Paper presented at the 11th World Wide Web Conference (WWW2002), Hawaii, USA.
- Norton, M. J. (2004, April 5, 2004). A Comparison between the JISC and Sakai Frameworks. 3rd. Retrieved May 30, 2006, from <http://sakaiproject.org/tech/S040405N.pdf>
- Nussbaumer, A., & Mistelbacher, A. (2002). XML ge-packt : XSLT, CSS und XSL-FO, XML-Schema, XPath, Xlink und XPointer (1. Aufl. ed.). Bonn: mitp.
- O'Reilly, T. (2006, June 24, 2006). What Is Web 2.0 Design Patterns and Business Models for the Next Generation of Software. Retrieved October 27, 2007, from <http://facweb.cti.depaul.edu/jnowotarski/se425/What%20Is%20Web%20point%20.pdf>
- OECD. (2007). *Giving Knowledge for Free: The emergence of open educational resources*: OECD, Centre for Educational Research and Innovation (CERI).
- Olivier, B., Roberts, T., & Blinco, K. (2005, July 2005). The eFramework for Education and Research: An Overview. R1. Retrieved June 30, 2007, from <http://www.e-framework.org/Portals/9/Resources/eframeworkV1.pdf>
- Olivier, B., & Tattersall, C. (2005). The Learning Design Specification. In R. Koper & C. Tattersall (Eds.), *Learning Design*

A Handbook on Modeling and Delivering Networked Education and Training (1 ed., pp. 21-40). Berlin Heidelberg: Springer. verlag.

Orlich, D. C. (2000). *Teaching Strategies: A Guide to Better Instruction* (6 ed.). New York: Houghton Mifflin Company.

OSI. (2007). The Open Source Definition. Retrieved February 06, 2007, from <http://www.opensource.org/docs/definition.php>

Palmér, M., Naeve, A., & Paulsson, F. (2004). *The SCAM Framework: Helping Semantic Web Applications to Store and Access Metadata*. Paper presented at the the European Semantic Web Symposium 2004, Heraclion Greece.

Papert, S. (1994). *Hur gör girafferna när den sover?* Göteborg: Daidalos AB.

Parrish, P. E. (2004). The Trouble with Learning Objects. *Educational Technology Research & Development (ETR & D)*, 52(1), 49-67.

Pask, G. (1960). Adaptive Teaching With Adaptive Machines. In A. A. Lumsdaine & R. Glaser (Eds.), *Teaching Machines and Programmed Learning: a source book* (Vol. 1, pp. 349-366). Washington, D. C.: National Education Association of the United States.

Patriarche, R., Diallo, G., Bernhard, D., Bassolet, C. G., Ferriol, S., & Gedzelman, S. (2005, 19-20 October). *A Tool for Textual and Conceptual Annotations of Documents*. Paper presented at the eChallenges 2005, Ljubljana, Slovenia.

Patrick, P. (2005, June 13-16). *Impact of SOA on enterprise information architectures*. Paper presented at the ACM SIGMOD international conference on Management of data, Baltimore, Maryland.

Paulsen, M. F. (2002, July 2002). Online Education Systems: Discussion and Definition of Terms Retrieved 2007-01-20, 2007, from <http://www.nettskolen.com/forskning/Definition%20of%20Terms.pdf>

Paulson, L. D. (2005). Building Rich Web Applications with Ajax. *Computer*, 38(10), 14-17.

Paulsson, F. (1998). Towards a Virtual Teacher College. Retrieved December 12, 2006, from <http://www.educ.umu.se/~frepa/presentations/1997/vtc.pdf>

Paulsson, F. (2002). Märkning av nationella styrdokument (pp. 1-8). Stockholm: Skolverket.

Paulsson, F. (2003). Standardized Content Archive Management – SCAM. *IEEE Learning Technology newsletter*, 5(1), 40-42.

Pavlov, I. P. (1927). *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex* (G. V. Anrep, Trans.). London: Oxford University Press.

Pedersen, J. (1998). *Informationsteknik i skolan. En forskningsöversikt*. (No. 98:343). Stockholm: Skolverket.

Pepper, S. (2000). *The Tao of Topic Maps: finding the way in the age of infoglut*. Paper presented at the XML Europe 2000, Paris, France.

Pepper, S., Vitali, F., Garshol, L. M., Gessa, N., & Presutti, V. (2005, 29 March, 2005). A Survey of RDF/Topic Maps Interoperability Proposals. Retrieved April 2, 2005, from <http://www.w3.org/TR/rdftm-survey/>

Piaget, J. (1968/2006). *Barnets själsliga utveckling*. Stockholm: Norstedts Akademiska Förlag.

- Piaget, J., & ebrary Inc. (1959/2002). *Judgment and reasoning in the child*. London: Routledge.
- Piaget, J., & Inhelder, B. (1997). *Selected works*.
- Piccoli, G., Ahmad, R., & Ives, B. (2001). Web-Based Virtual Learning Environments: A Research Framework and a Preliminary Assessment of Effectiveness in Basic IT Skills Training. *MIS Quarterly*, 25(4), 401-426.
- Polsani, P. R. (2003). Use and Abuse of Reusable Learning Objects. *Journal of digital information*, 3(4).
- Powell, A., Nilsson, M., Naeve, A., Johnston, P., & Baker, T. (2004, 4 June 2007). DCMI Abstract Model. Retrieved 10 February, 2008, from <http://dublincore.org/documents/abstract-model>
- Pressey, S. L. (1927). A machine for automatic teaching of drill material. *School and Society*, 25(645), 549-552.
- Psyche, V., Bordeau, J., Nkambou, R., & Mizoguchi, R. (2005). *Making Learning Design Standards Work with an Ontology of Educational Theories*. Paper presented at the 12th International Conference on Artificial Intelligence in Education, Amsterdam.
- Quinn, C., & Hobbs, S. (2000). Learning Objects and Instruction Components. *Educational Technology & Society*, 3(2).
- Rawlings, A., Koper, R., van Rosmalen, P., Rodríguez-Artacho, M., & Lefrere, P. (2002). *Survey of Educational Modeling Languages (EMLs)*.
- Rehak, D., & Mason, R. (2003). Keeping Learning in Learning Objects, in the Special Issue on Reusing Online Resources: A Sustainable Approach to eLearning. *Journal of Interactive Media in Education*, 2003(1), 22-27.
- Reigeluth, C. M. (1983). *Instructional-design theories and models: an overview of their current status* (Vol. 1). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Reigeluth, C. M. (1999). *A new paradigm of instructional theory* (1 ed. Vol. 2). Mahwah, N.J. : Lawrence Erlbaum.
- Reigeluth, C. M. (1999a). *Instructional-Design Theories and Models: A New Paradigm of Instructional Theory* (Vol. II). Mahwah, NJ: LEA.
- Reigeluth, C. M. (1999b). The Elaboration Theory: Guidance for Scope and Sequence Decisions. In C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models: A New Paradigm of Instructional Theory* (Vol. 2, pp. 425-453). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
- Reigeluth, C. M. (1999c). What is Instructional-Design Theory and how is it changing? In C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models: A New Paradigm of Instructional Theory* (Vol. 2, pp. 5-31). Mahwah, NJ: LEA.
- Reigeluth, C. M. (2008). *Instructional-design theories and models, Volume III* (1 ed. Vol. 3). Mahwah, N.J. : Routledge.
- Richter, C., Nejd, W., & Allert, H. (2002). *Learning Objects on the Semantic Web. Explicitly Modelling Instructional Theories and Paradigms*. Paper presented at the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, Chesapeake, VA.

- Rodriguez, M. A., & Egenhofer, M. J. (2003). Determining Semantic Similarity Among Entity Classes from Different Ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 15(2), 442-456.
- Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., et al. (2005). Web Service Modeling Ontology. *Applied Ontology*, 1(1), 77-106.
- Saettler, P. (2004). *The Evolution of Educational American Technology*. Connecticut: Information Age Publishing.
- Salmenjoki, K., Paulsson, F., & Vainio, A. (2007, 6-8 September). *Social Learning - How Web is Changing the Way We Process Information and Knowledge*. Paper presented at the 5th International Conference on Emerging e-learning Technologies and Applications (ICETA), Stará Lesná, Slovak Republic.
- Schank, R. C., Berman, T. R., & Macpherson, K. A. (1999). Learning by Doing. In C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models: A New Paradigm of Instructional Theory* (Vol. 2, pp. 161-181). Mahwa, NJ: LEA.
- Schluep, S., Bettino, M., & Guttormsen Schär, S. (2005, March 3-4, 2005). Modularization and Structured Markup for Learning Content in an Academic Environment. *ProLearn-iClass Thematic Workshop on Learning Objects in context*. Retrieved August 27, 2005, from <http://www.cs.kuleuven.ac.be/~hmdb/ProlearnIClass/papers/Schluep.pdf>
- Schult, L. Bloom's Taxonomy. Retrieved October 10, 2007, from http://www.odu.edu/educ/lischult/blooms_taxonomy.htm
- Schunk, D. H. (1991). *Learning Theories: An Educational Perspective* (1 ed.). New York: Macmillan.
- Schunk, D. H. (2003). *Learning Theories An Educational Perspective* (4 ed.). New Jersey: Pearson Higher Education.
- Seels, & Richey. (1994). *Instructional technology: The definition and domains of the field*. Bloomington, IN: Association for Educational Communications and Technology.
- Sessions, R. (2001, March 28, 2001). Java 2 Enterprise Edition (J2EE) versus The .NET Platform Two Visions for eBusiness. Retrieved October 01, 2007, from <http://www.objectwatch.com/FinalJ2EeandDotNet.doc>
- Shadbolt, N., Hall, W., & Berners-Lee, T. (2006). The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3), 96-101.
- Shaw, R. (2005, December 07). Web 2.0? It doesn't exist. Retrieved February 16, 2007, from <http://blogs.zdnet.com/ip-telephony/?p=805>
- Shuell, T. J. (1990). Phases of meaningful learning. *Review of Educational Research*, 60(4), 531-547.
- Silveira, I. F., Fernando de Araújo Jr., C., Amaral, L. H., Alcântara de Oliveira, I. C., Schimiguel, J., Ledón, F.-P., Manuel, et al. (2007). Granularity and Reusability of Learning Objects. In A. Koohang & K. Harman (Eds.), *Learning Objects and Instructional Design* (pp. 139-170). Santa Rosa, CA: Informing Science Press.
- Sim, Y. W., Wang, C., Carr, L. A., Davis, H. C., Gilbert, L., Grange, S., et al. (2005, 19-22 September). *A Web/Grid Services Approach for a Virtual Research Environment Implementation*. Paper presented at the Fourth e-Science All Hands Meeting (AHM 2005), Nottingham, UK.

- Sim, Y. W., Wang, C., Gilbert, L., & Wills, G. B. (2005). *An Overview of Service-Oriented Architecture* (Technical Report No. ECSTR-IAM05-004). Southampton: Electronics and Computer Science, University of Southampton.
- Simpson, T. L. (2002). Dare I oppose constructivist theory? *The Educational Forum*(66), 347-354.
- Skinner, B. F. (1958). Teaching Machines. *Reprinted from Science*, 128 (3330), 969-977.
- Skinner, B. F. (1960). Teaching Machines. In A. A. Lumsdaine & R. Glaser (Eds.), *Teaching Machines and Programmed Learning: a source book* (Vol. 1, pp. 137-158). Washington, D. C.: National Education Association of the United States.
- Skinner, B. F. (1976). *About Behaviourism*. New York: Vintage Books.
- Smith, J. (1999, October 29, 1999). Browser Compatibility Chart. Retrieved February 12, 2008, from <http://labnt1.bizlab.mtsu.edu/Lab/tectrain/BrowserCompatibility.htm>
- Snelbecker, G. E. (1974). *Learning theory, instructional theory, and psychoeducational design*. New York.
- Snelbecker, G. E. (1999). Current Progress, Historical Perspective, and Some Task of the Future of Instructional Design Theory. In C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models: A New Paradigm of Instructional Theory* (Vol. 2, pp. 653-674). Mahwah, NJ: LEA.
- Song, W. W., & Andersson, M. B. (1999, October 24-30). *A Component-based Framework for Description and Management of Learning Objects*. Paper presented at the WebNet 99 World Conference on the WWW and Internet, Honolulu, Hawaii.
- Song, W. W., Andersson, M. B., Hakkarainen, S. E., Karlberg, P., & Soltesz, T. (1999). Metadata for the management of electronic documents in the governmental organizations and learning objects in the learning domain (Research report No. SITI 99:03). Kista: SITI, SISU.
- Sosteric, M., & Hesemeier, S. (2002). When is a Learning Object not an Object: A first step towards a theory of Learning objects. *International Review of Research in Open and Distance Learning*, 3(2).
- Subramaniam, M. A. (2002, Summer 2002). Garden-Based Learning in Basic Education: A Historical Review. *Monograph (Focus)*. Retrieved September 13, 2007, from <http://cyd.ucdavis.edu/publications/monograph.html>
- Szyperski, C. (2002a). *Component Software - Beyond Object-Oriented Programming* (2 ed. Vol. 1). New York: ACM Press.
- Szyperski, C. (2002b). Object and component "wiring" standards. In *Component Software - Beyond Object-Oriented Programming* (2 ed., Vol. 1, pp. 207-230). New York: ACM Press.
- Szyperski, C. (2002c). What a component is and is not. In *Component Software - Beyond Object-Oriented Programming* (2 ed., Vol. 1, pp. 35-48). New York: ACM Press.
- Szyperski, C., & Pfister, C. (1997). Workshop on Component-Oriented programming, Summary. *Special Issues in Object-Oriented Programming - ECOOP '96*.
- Söderlund, A. (2000). Det långa mötet. IT och Skolan. Om spridning och anammande av IT i den Svenska skolan. Centrum för forskning i lärande: Luleå tekniska universitet.
- Tennyson, R., Schott, F., Seel, N., & Dijkstra, S. (1997). *Instructional design: International perspectives*. Mahwah, NJ: Erlbaum.

- Thorne, S., & Kahn, J. (2003). *O.K.I.™ Architectural Concepts* (Technical report (draft)): Massachusetts Institute of Technology.
- Trigg, R. H., & Irish, P. M. (1987, November 13-15). *Hypertext Habitats: Experiences of Writers in NoteCards*. Paper presented at the Hypertext'87, Chapel Hill, North Carolina, USA.
- Walker, S. F. (1990). A brief history of connectionism and its psychological implications. *AI & Society*, 4(1), 17-38.
- Van Assche, F., & Vuorikari, R. (2006). A Framework for Quality of Learning Resources. In U.-E. Ehlers & J. P. Pawlowski (Eds.), *European Handbook for Quality* (pp. 443-456): Springer Berlin Heidelberg.
- Warren, P. (2006). Knowledge Management and the Semantic Web: From Scenario to Technology. *IEEE Intelligent systems*, 21(1), 53-59.
- Varsha, A., Hansen, T., & Reeser, P. (2002). Performance comparison of dynamic web platforms. *Computer Communications*, 26(8), 888-898.
- Watson, J. B. (1925). *Behaviorism*. New York: The People's Institute Publishing Co.
- Watson, J. B., & Rayner, R. (1920). CONDITIONED EMOTIONAL REACTIONS. *Journal of Experimental Psychology*, 3(1), 1-14.
- Weinreich, R., & Sametinger, J. (2001). Component Models and Component Services: Concepts and Principles. In G. T. Heinemann & W. T. Council (Eds.), *Component-Based Software Engineering: putting the pieces together* (pp. 33-48). Upper Saddle River, NH: Addison-Wesley.
- Weller, M. j., Pegler, C. A., & Mason, R. D. (2003, 9-12 February). *Putting the pieces together: What working with learning objects means for the educator*. Paper presented at the eLearn International, Edinburgh, Scotland.
- Ver, M. V., E, M., Dominigue, J., Lanzoni, M., A, S., & Ciravegna, F. (2003, July 22-26). *Ontology Driven Tool for Semantic Markup*. Paper presented at the ECAI 2002 Workshop on Semantic Authoring, Annotation & Knowledge Markup, Lyon.
- Wikipedia. (2007, April 05, 2007). Dynamic web page - Wikipedia, the free encyclopedia. Retrieved April 09, 2007, from http://en.wikipedia.org/wiki/Dynamic_Web_pages
- Wiley, D. A. (1999). The Post-LEGO Learning Object. Retrieved 16 March, 2002, from <http://wiley.ed.usu.edu/docs/post-lego.pdf>
- Wiley, D. A. (2000). *Learning Object Design and Sequencing Theory*. Brigham Young University.
- Wiley, D. A. (2002a). 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* (First ed., pp. 3-23). Bloomington: Agency for Instructional Technology and Association for Educational Communications & Technology.
- Wiley, D. A. (2002b). *The Instructional Use of Learning Objects* (First ed.). Bloomington: Agency for Instructional Technology and Association for Educational Communications & Technology.
- Wiley, D. A. (2003). Learning Objects: Difficulties and opportunities. Retrieved 20 February, 2008, from http://wiley.ed.usu.edu/docs/lo_do.pdf
- Wiley, D. A. (2007). *On the Sustainability of Open Educational Resource Initiatives in Higher Education*: OECD, Centre for Educational Research and Innovation (CERI).

- Williams, J. (2001). The Business Case for Components. In G. T. Heinemann & W. T. Council (Eds.), *Component-Based Software Engineering: Putting the Pieces Together* (1 ed., Vol. 1, pp. 79-97): Addison-Wesley.
- Wilson, S., Blinco, K., & Rehak, D. (2004, July, 2004). *An e-Learning Framework A Summary*. Paper presented at the Alt-I-lab 2004.
- Wilson, S., Olivier, B., Jeyes, S., Powell, A., & Franklin, T. (2004). A Technical Framework to Support e-Learning. Retrieved August 28, 2005, from http://www.jisc.ac.uk/uploaded_documents/Technical%20Framework%20feb04.doc
- Wilson, S., & Rees Jones, P. (2005). What Is...IMS Learner Information Packaging? *CETIS briefings on e-learning standards series*. Retrieved October 05, 2005, from <http://www.cetis.ac.uk/lib/media/lipbrief.pdf>
- Voss, J. (2007, July 18-24). *Tagging, Folksonomy & Co - Renaissance of Manual Indexing?* Paper presented at the 10th International Symposium of Information Science, Cologne, Germany.
- Vygotsky, L. (1978). *Mind in society: the development of higher psychological process*. Cambridge: Harvard University Press.
- Yoder, J., & Barcalow, J. (1997, September 3-5). *Architectural Patterns for Enabling Application Security*. Paper presented at the 4th Pattern Languages of Programming Conference, Monticello, Illinois, USA.
- Yu, L. (2007). *Introduction To the Semantic Web and Semantic Web Services* (1 ed.). Boca Rayton, FL: Taylor & Fancis Group.
- Zaihrayen, I., Sun, L., Giunchiglia, F., Pan, W., Ju, Q., Chi, M., et al. (2007, November 2007). *From Web Directories to Ontologies: Natural Language Processing Challenges*. Paper presented at the 6th International Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea.

Appendix A – List of acronyms

This appendix contains a list of the acronyms that are most frequently used in this thesis. The acronyms are only briefly described in the table since most of them are explained and discussed in detail in chapter 3 and 5.

Acronym	Concept	Short description
AJAX	Asynchronous JavaScript and XML	A collection of technologies and principles for dynamic user interfaces on the web
CBI	Computer-Based Instruction	Instructional concept using computers.
CBT	Computer-Based Training	Instructional concept using computers.
CGI	Common Gateway Interface	A standard for interfacing with web servers.
DTD	Document Type Definition	A schema language for XML and SGML.
EML	Educational Modeling Language	A collective term for metalanguages that describe pedagogical process.
FOAF	Friend of a Friend	A schema for representing people and social networks.
HTML	Hypertext Markup Language	The original markup language of the web.
HTTP	Hypertext Transfer Protocol	A communication protocol for the Internet and the web
LMS	Learning Management System	Software system for managing courses and learning content.
LOM	Learning Object Metadata	A metadata model for describing learning objects.
LOR	Learning Object Repository	Repository for learning resources and metadata.
OKI	Open Knowledge Initiative	The initiative behind the OSID.
OSID	Open Service Interface Definitions	OKI's Software (service) interfaces for SOA.
OWL	Web Ontology Language	An ontology markup language.
RDF	Resource Description Framework	A framework for representing data on the Semantic Web.
RDFS	RDF Schema	Provides basic elements for describing ontologies with RDF.
REST	Representational state transfer	An architecture style that utilizes the characteristics of the web.

RMI	Remote method Invocation	A Java technology for remote invocations.
RSS	Really Simple Syndication, Rich Site Summary or RDF Site Summary	Formats for syndicating content.
SCORM	Sharable Content Object Reference Model	A reference model for sharable content and LMSes.
SGML	Standard Generalized Markup Language	A metalanguage for defining markup languages.
SOA	Service Oriented Architecture	An architectural model for loosely coupled services.
SOAP	Simple Object Access Protocol	Protocol for exchanging XML-based messages over the Internet.
TEI	Text Encoding Initiative	A markuplanguage for digital texts.
TEL	Technology Enhanced Learning	Learning that are enhanced using ICT.
UDDI	Universal Description, Discovery and Integration	A Web Service registry.
VLE	Virtual Learning Environment	General term for ICT-based learning environments.
W3C	The World Wide Web Consortium	A consortium that promotes interoperable technologies for the web.
WBL	Web-Based Learning	Education and learning distributed on the web.
WBT	Web-Based Training	Education and training distributed on the web.
WSDL	Web Service Description Language	A description language for web services.
XML	Extensible Markup Language	A standard for defining markup languages.
XML-RPC	XML Remote Procedure Call	An XML-based protocol for remote calls over http.
XSLT	Extensible Stylesheet Language Transformations	A language for transforming XML documents.

Appendix B – Appended papers

This appendix contains the six papers that are appended to this thesis