Master Thesis

Reference Architecture for e-Learning Solutions

Job Habraken
Student number: 835939573

Open University
Faculty Computer Science
January 28, 2008
Institute
Open University
Faculty Computer Science

Supervising committee
Open University
Chairmen: Name: Prof. dr. J.T. (Johan) Jeuring
E-mail: johan.jeuring@ou.nl

Secretary: Name: Ir. S. (Sylvia) Stuurman
E-mail: sylvia.stuurman@ou.nl

External referee
University of Groningen: Name: P (Paris) Avgeriou, PhD
E-Mail: paris@cs.rug.nl

Delegate from BT
BT: Name: J.C. (Joep) de Jong
Role: Director of BT Learning Solutions
E-mail: joep.dejong@bt.com
It is not necessarily complicated.
It is not necessarily simple.

Christopher Alexander, In ‘The Timeless Way of Building’
Acknowledgements

I have been a student of the Open University for a very long time. At first I only studied topics out of interest and topics that could be of use for my job. My way of studying at the Open University changed for several reasons. The first reason was because BT Learning Solutions, the business unit I am working for, changed it focus from providing specialist instructor led training to developing e-Learning solutions for customers. A second reason was because my role changed from a team leader of specialist trainers to a manager of a development team. The last reason was because the Open University implemented the Bachelors-Masters structure for the faculty of Computer Science. This restructuring of the faculty at the Open University gave me the opportunity to complete my Computer Science study in two phases. It also gave me the opportunity to start studying al lot of new topics such as Design Patterns, Distributed Programming, ICT Architecture and Software Management. That’s where my interest in the usage of patterns and software architectures started. My colleagues of the development team did the rest; they influenced my research interest for patterns, pattern languages and (reference) architectures, most probably without knowing.

I would like to thank Johan Jeuring and Sylvia Stuurman from the Open University for all their support and comments on the different sections of my thesis. I would like to thank Paris Avgeriou, as external referee, for his insightful and fundamental comments on my thesis. Besides being the external referee for my thesis his articles on architectures, quality attributes, reference architectures, patterns and pattern languages greatly influenced my research interest in these topics.

Last but not least, I would like to thanks my wife Ria for her patience and support, she knew what it takes to write a thesis. Hopefully we will have some more spare time to spend together.

Job Habraken, 2008
Table of content

1 SUMMARY.............................................................................................................................. 3
   1.1 Summary ......................................................................................................................... 3
   1.2 Samenvatting ................................................................................................................ 3

2 INTRODUCTION .................................................................................................................. 3
   2.1 E-learning solutions........................................................................................................ 3
   2.2 Reference architecture ................................................................................................. 3
   2.3 Research questions ........................................................................................................ 3
   2.4 Approach ....................................................................................................................... 3
   2.5 Thesis outline ................................................................................................................ 3

3 THEORETICAL UNDERPINNINGS FOR E-LEARNING SOLUTIONS....................................... 3
   3.1 Distributed versus integrated solutions ......................................................................... 3
   3.2 Reference model for e-Learning solutions .................................................................... 3
      3.2.1 Introduction .............................................................................................................. 3
      3.2.2 UKeU e-Learning framework .................................................................................. 3
      3.2.3 A model for Web-based Instructional Systems ....................................................... 3
      3.2.4 LTSA: IEEE P1484-1 Standard ............................................................................. 3
      3.2.5 Analysis of the WbIS model, UKeU framework and the LTSA ................................ 3
   3.3 Documenting a reference architecture for e-Learning solutions .................................... 3
      3.3.1 Introduction .............................................................................................................. 3
      3.3.2 Reference models, reference architectures and software architectures .................. 3
      3.3.3 Documentation package for a reference architecture ................................................. 3
   3.4 Architectural Viewpoints, Views and Patterns................................................................. 3
      3.4.1 Introduction .............................................................................................................. 3
      3.4.2 Architectural Viewpoints ......................................................................................... 3
      3.4.3 Architectural Views ................................................................................................ 3
      3.4.4 Patterns, pattern stories and a pattern language for e-Learning solutions ............... 3

4 REFERENCE ARCHITECTURE FOR E-LEARNING SOLUTIONS........................................... 3
   4.1 Business case for a reference architecture for e-Learning solutions ............................. 3
      4.1.1 Introduction .............................................................................................................. 3
      4.1.2 Domain Model for an e-Learning solution ............................................................... 3
      4.1.3 Rationale for creating a reference architecture ....................................................... 3
      4.1.4 Benefits of a reference architecture for e-Learning solutions ................................ 3
      4.1.5 Scope of the reference architecture ....................................................................... 3
   4.2 Stakeholders and their concerns .................................................................................... 3
      4.2.1 Introduction .............................................................................................................. 3
      4.2.2 Stakeholders ............................................................................................................ 3
      4.2.3 Concerns .................................................................................................................. 3
   4.3 Functional requirements, quality attributes and implementation constraints ............... 3
1 Summary

1.1 Summary

BT Learning Solutions (BT LS) is a business unit of British Telecomm that focuses on creating e-Learning solutions for its customers. A lot of research has been done in the area of models and frameworks for e-Learning solutions with a focus on the educational area. What is lacking is a reference architecture in which learning technology components from different suppliers are integrated into one e-Learning solution for a customer. A reference architecture describes the essence of a software architecture and the most significant and relevant aspects.

In this thesis three models and frameworks for e-Learning solutions: the UKeU e-Learning framework, a model for Web-based Instructional System developed by Retalis and Avgeriou and the LTSA developed by the IEEE are discussed. These three models are the foundation for a Managed Learning Environment that serves as the Domain Model for e-Learning solutions.

The theoretical underpinnings of reference architectures, architectural patterns and pattern languages are also discussed in this thesis. These theoretical underpinnings are used to outline what should be part of a documentation package for a reference architecture. The reference architecture for e-Learning solutions will be developed by re-using architectural patterns from different domains and is evaluated by a list of quality attributes.

Based on the theoretical underpinnings that are discussed in this thesis, a pattern language for e-Learning solutions will be proposed that groups together the architectural patterns that are used to design the views of the reference architecture.

Two business cases from BT LS will be used to instantiate and evaluate the reference architecture. The two instantiations demonstrate that the reference architecture could be used by the development team of BT LS to design architectures for e-Learning solutions.
1.2 Samenvatting

BT Learning Solutions (BT LS) is een business unit van British Telecom die zich gespecialiseerd heeft in het ontwikkelen van e-Learning oplossingen voor haar klanten. Veel onderzoek is gedaan op het gebied van modellen en raamwerken voor e-Learning oplossingen met een focus op de educatie markt. Wat mist is een referentie architectuur waarin leertecnologieën van verschillende leveranciers geïntegreerd zijn in een e-Learning oplossing voor een klant. Een referentie architectuur beschrijft de essentie van een software architectuur en de meeste significante en relevante aspecten.

In deze scriptie worden de volgende drie modellen en raamwerken voor e-Learning besproken: het e-Learning raamwerk ontwikkeld voor de UKeU, het model voor Web-based Instructional Systems ontwikkeld door Retalis en Avgeriou en het LTSA ontwikkeld door het IEEE. Deze drie modellen vormen de theoretische basis voor een Managed Learning Environment die als domeinmodel dienen voor e-Learning oplossingen.

De theorie van referentie architecturen, patronen en patroontalen zal worden besproken in deze scriptie en gebruikt worden om aan te geven waaruit de documentatie van een referentie architectuur zal moeten bestaan. De referentie architectuur zal worden ontwikkeld op basis van het hergebruik van patronen uit de verschillende onderzoeksdomeinen en zal worden geëvalueerd aan de hand van een lijst kwaliteitsattributen.

Op basis van de theoretische beschouwingen van deze scriptie wordt een patroontaal voor e-Learning oplossingen voorgesteld waarin de patronen die gebruikt zijn voor de ontwikkeling van de views van de referentie architectuur gegroepeerd zijn.

Twee business cases van BT LS zullen als voorbeeld gebruikt worden om de referentie architectuur te evalueren. De twee voorbeelden tonen aan dat de referentie architectuur als basis kan dienen voor toekomstige nieuw te ontwikkelen architecturen voor e-Learning oplossingen door het ontwikkelteam van BT LS.
2 Introduction

The business unit Learning Solutions of British Telecom (BT LS) has been developing and implementing learning solutions for their customers for more than a decade. Over the last 10 years the focus of BT LS has shifted from mainly traditional classroom based training solutions to learning solutions in which web based learning technologies play an important role. This thesis presents a reference architecture for e-Learning solutions, which the BT LS development team can use to develop new e-Learning solutions for their customers.

This section presents the background of this thesis. Section 2.1 provides information about e-Learning solutions. Section 2.2 introduces the concepts of software architectures, reference architectures and patterns. Section 2.3 brings both areas together in the problem statement and the research questions of this thesis. Section 2.4 describes the approach of how the research questions have been addressed. Section 2.5 gives an overview of the outline of this thesis.

2.1 E-learning solutions

E-Learning always excites much debate, hype and anxiety amongst educators, trainers, training departments, etc. A complex diagram could be drawn with e-Learning intersecting with other topics in education and training such as life long learning, performance management, talent management etc. Besides these areas there is also an overlap with broader technological developments such as web services, integration styles or educational technological standards ([Ros06], [Wel07]). Different terminologies have been used to define e-Learning, this makes it difficult to give a generic definition. Rosenberg makes a distinction between traditional e-Learning and smart enterprise e-Learning. Traditional e-Learning focuses on delivering online courses and has instructional design as the default and main approach. Smart enterprise e-Learning is integrated in the business processes and focuses on delivering instruction and information and enables collaboration [Ros06]. Weller describes “e-Learning for the enthusiasts” as e-Learning where the internet facilitates a two-way communication and encourages discussion, dialogue and community that is not limited by time and place. The role of the educator is to facilitate dialogue and support of students in their understanding of resources [Wel07]. Paulsen defines e-Learning as interactive learning in which the learning content is available online and which also provides feedback on the students’ activities [Pau02]. All these definitions have in common that the internet or intranet is used as a medium to provide learning services and that learners should be able to receive feedback on their learning. What is lacking in these definitions is the administration and assessment part of learning. E-Learning in this thesis is defined as “Learning where the internet or intranet plays an important role in the delivery, support, administration and assessment of learning” [JMK04].

The political, technical and economical arguments that surround e-Learning are all reflected in the choice, deployment and development of a Managed Learning Environment (MLE) in an organisation. The Joint Information Systems Committee (JISC) has defined an MLE as a “System that uses technology to enhance and make more effective the network of relationships between learners, teachers and organisers of learning through integrated support for richer communication and activities” [JISC04]. This definition is also adopted by other educational institutions [JISC05]. In this thesis the focus is on organisations in general and not on educational institutions. Therefore the following definition of an MLE is used: “The whole range of information systems and processes of an organisation that contribute directly, or indirectly, to learning and the management of that learning.” ([Wel07], [WJOP04]).
An e-Learning solution, in this thesis, is defined as a solution in which a Managed Learning Environment (MLE) is developed that comprises different learning technology applications such as an LMS or an LCMS that contribute to learning and the management of that learning. An e-Learning solution, as defined in this section, is a very broad and multi-faceted topic and therefore its scope must be narrowed to fit within the timeframe available for a Master thesis. This thesis focuses on the integration aspects of different learning technology applications that form part of an MLE.

2.2 Reference architecture

A software architecture is defined in the literature “As the structure or structures of a system, which comprises software elements, the externally visible properties of those elements, and the relation between them” ([BCK03], [CBB+05]).

In all domains, including the e-Learning domain, two simultaneous trends are identified [Mul07]:

- Increase of the complexity, scope and size of solutions;
- Increasing dynamics such as requests for shorter time to market of solutions and more interoperability between systems that form part of a solution.

These trends cause a transition from a “simple” closed system to a solution with distributed open systems. Many of those distributed open systems are developed at multiple locations, by multiple suppliers for multiple organisations. A reference architecture is a reaction to the two trends and describes the essence of the software architecture and the most significant and relevant aspects. The purpose of a reference architecture is to provide guidance for the development of new versions of architectures for systems ([Ree02], [Mul07]).

The benefits of a reference architecture for organisations are [WOJP04]:

- **A reference architecture provides better return on technology investment.** New learning technology components can be developed or acquired when needed, which means that only those parts of an e-Learning solution that really need to be changed are replaced, retaining the other components that comprise the solution. By doing so purchasing and implementation costs are reduced.

- **A reference architecture enables faster deployment of technology.** As learning technology components are independent it will be easier to deploy new components as long as the new components are compatible with the existing interfaces. If the latter is not the case it may still be simpler to alter or to replace other learning technology components to supply the requirements of the e-Learning solution.

- **A reference architecture provides a modular and flexible technology base.** The rationale for the reference architecture is to enable the development of flexible e-Learning solutions, in which individual learning technology components can be added or replaced more easily than in a solution without it.

An important step in creating a software architecture or a reference architecture is its design. In this thesis patterns are used to design the relevant parts of the reference architecture. A popular description of patterns is: “a pattern is a solution to a recurring problem in a context”. Patterns rarely exist in isolation; each pattern generally has one or more relationships with other patterns ([Ris98], [BHS07b]). A collection of patterns can be collected in a pattern language for a particular purpose such as presenting patterns within a particular architecture, so that it can be organised into problem areas that characterise the architecture [BHS07b].
2.3 Research questions

The BT LS business unit develops and implements e-Learning solutions for their customers. The development team of BT LS does not have a set of architectural patterns nor a reference architecture that they can re-use to design architectures that are the foundation for its e-Learning solutions. This leads to the following problem statement.

Problem statement:

This thesis investigates if a reference architecture for e-Learning solutions can be developed and its views can be designed by re-using architectural patterns. This thesis also investigates if a pattern language can be developed for e-Learning solutions.

Research questions that are addressed in this thesis:

1. Is it possible to develop a pattern language to design a reference architecture for e-Learning solutions by re-using patterns from the different domains?
2. How is this pattern language used to design this reference architecture?
3. How can two architectures, from business cases of BT LS, be instantiated from the reference architecture and how are these architectures evaluated?
4. How is the reference architecture extended with specific implementation details for each of the business cases to make these implementations conform to this reference architecture?

The two business cases are examples of architectures that are developed and implemented by BT LS. Both business cases contain sensitive customer information that is not publicly available. The architectures of both business cases are therefore added as an appendix at the end of this thesis. These appendices are available for the thesis committee and for the BT LS business unit. The two business cases are referenced to as case A and case B throughout this thesis.

2.4 Approach

To answer the research questions four main objectives are addressed in this thesis:

1. Select and describe patterns from different domains that can be used to design the relevant parts of a reference architecture for an e-Learning solution.
2. Develop a reference architecture for e-Learning solutions.
3. Develop a pattern language that contains the patterns that are used to design the reference architecture.
4. Validate the two architectures that instantiate the reference architecture and show how the reference architecture can be extended with specific implementation details of each of the business cases.

2.5 Thesis outline

The outline of the thesis is structured as follows. Section 3 describes the theoretical background of the reference models for e-Learning solutions that are used in this thesis. Section 3 also discusses the relevant parts of a reference architecture and the concepts of architectural views, view-points and patterns. Section 4 addresses the first two objectives namely the development of a reference architecture for e-Learning solutions and the usage of patterns for its design. The third objective is discussed in section 5 where an overview is provided of the relevant architectural patterns in an e-Learning context and a pattern language is proposed for e-Learning solutions. Section 6 revisits the answers on the research questions, discusses the problem statement and provides recommendations for future work.
Appendix A provides a list with acronyms and a glossary with a description of the most important terms used in this thesis. Appendix B and C describe the architectures of the two business cases that validates the business cases and address the last objective of the thesis.
3 Theoretical underpinnings for E-Learning Solutions

3.1 Distributed versus integrated solutions

An MLE comprises several information systems that all contribute directly or indirectly to learning or learning management of an organisation. These information systems do not operate as stand alone systems but are integrated with each other and could be distributed across the network of an organisation. This section describes the challenges that integrated solutions face and the four fundamental approaches to integrate these. These four approaches are used to design the component interaction views in section 4.5.5.

An MLE could be implemented in a distributed environment in which its components run by themselves and are integrated with one another in a loosely coupled way. This enables each component to focus on its own functionalities and responsibilities and connect to other components for related functionality. So for instance a Learning Management System (LMS) focuses on functionality such as the enrolment of users onto learning activities but it delegates the actual delivery of the content to a Learning Content Management System (LCMS). Integrated components communicate asynchronously and don’t have to wait for a response and can proceed without a response to perform other tasks concurrently until the response is available.

Integrated solutions have to deal with a few fundamental challenges [HOWO04]:

- Networks are unreliable. Integrated solutions have to transport data from one server to another server across networks. By distributing servers across networks, an architecture has to deal with a large set of possible problems such as delays and interruptions on the network.
- Networks are slow. Sending data across a network will be slower compared with local method calls.
- Any two applications are different. An integrated solution has to deal with different technologies using different interface for communication with another.
- Change is inevitable. A solution has to be able to cope with changes. An integrated solution has to keep pace with all the applications it connects. An integrated solution should therefore avoid a tight coupling with the applications it connects. A change to solution with tightly connections between applications could have an impact on the whole integrated solution.

To overcome these challenges developers have created four main approaches for integrating applications [HOWO04]:

- File transfer: One application generates a file which is sent across the network to another application. The applications need to agree on filename, location, file format, scheduling of when it will be read and created, and who will delete the file.
- Shared database: Multiple applications are using the same database schema, located in a single physical database. No data-transfer is required in this approach.
- Remote Procedure Invocation: One application exposes some of his functionality to other applications so that it can be accessed remotely by other applications.
- Messaging: One application publishes messages which are consumed by other applications.

All four approaches solve integration problems, where each of the styles has its distinct advantages and disadvantages. Buschman et al. also identify Publisher-Subscriber as an integration style between components.
In this thesis components, applications and systems are considered to be interchangeable. Components, applications and systems are referred to as runtime entities that can be deployed independently and are subject to composition by third parties.

3.2 Reference model for e-Learning solutions

3.2.1 Introduction

This section describes three models that are used as a basis for a reference model for e-Learning solutions. The first model is the e-Learning framework from UK e-Universities Worldwide (UKeU) [UKEU02]. This framework was developed and tested by the UKeU in conjunction with SUN’s Global Education and Research Group. The second model is the model for Web-based Instructional Systems (WbIS). This model is described by Retalis and Avgeriou in [REAV02]. The third model is the Learning Technology Systems Architecture (LTSA) developed by IEEE [IEEE02]. All three models are used in this thesis and form the basis for Domain Model of an MLE.

Subsection 3.2.2 gives an overview of the UKeU framework, subsection 3.2.3 describes the model from Retalis and Avgeriou and subsection 3.2.4 gives an overview of the LTSA. Subsection 3.2.5 analyses the three models, and discusses how the models will be used in the thesis.

3.2.2 UKeU e-Learning framework

The UKeU is a framework that is based on open standards and allows a complete separation of the different components from vendor implementations. This allows companies to pick and choose the best of breed solutions for each of the components and how the components are integrated in an e-Learning solution ([UKEU02], [SUN03]).

Figure 3.2-1 gives an overview of the UKeU e-Learning framework.

As displayed in figure 3.2-1, the framework is built using 4 layers. Each layer describes the services that perform specific tasks in a certain area of the e-Learning solution. The 4 layers are described as follows:

- **Layer 1: Portal**
  The portal provides a single port of entry for the users to the different services in the e-Learning framework and is responsible for the authentication of the users. Access is based on a user-profile.

- **Layer 2: Common Services**
  These are the services everyone needs; they are not tied to any particular pedagogic function and should therefore not be integrated inseparably with one of the pedagogic items. The model makes a distinction between three services: User Management (service that authorise users and assigns privileges to every user and provides a
lifelong unique-id for users), Collaboration (service that provides synchronous and asynchronous interfaces between all users) and Event Management (services that provides calendar, scheduling and reminder functions for users).

- **Layer 3: Learning Services**
  These services provide the core functionality for e-Learning solutions. This layer determines four different services:
  
  - Learning Content Management System (LCMS) is a service which allows authors to register, store, assemble, manage, package and publish learning content.
  
  - Learning Management System (LMS) is a service which registers users, tracks courses in the catalogue, records data from learners and provides searching and reporting capabilities to users of the service.
  
  - Assessment System is a service which measures students’ performance against specific goals using a variety of tools.
  
  - Learning Administration System is a service which acts as an interface to other information systems such as HR-systems.

- **Layer 4: Database(s)**
  Databases used by the different components in this framework.

The framework as suggested by UKeU and SUN is similar to the more granular technical framework to support e-Learning from Wilson et al. [WOJP04]. In this thesis the focus is on creating a reference architecture for an MLE that integrates several Learning Technology Components described in the UkeU framework. Each of the MLE components provides a set of services as described by the framework of Wilson et al. An LMS for instance provides features such as course management, cataloguing, group management, scheduling, resource management or resource discovery whereas an LCMS provides features such as sequencing, packaging, activity authoring, content management ([SUN03], [UKEU02], [WOJP04]).

3.2.3 A model for Web-based Instructional Systems

A Web-based Instructional System (WbIS) is an instructional system that incorporates organisational, administrative, instructional and technological components to support and (partially) automate an instructional process in a particular area [REAV02]. Retalis and Avgeriou describe a model for a WbIS by dividing the system into three subsystems. Each of the subsystems makes clear how users, learning resources and the technical infrastructure should work together in the instructional process. The model focuses on the technical part and does not represent theories of learning.

Retalis and Avgeriou describe two steps to come to an architecture for a WbIS. The first step is to define a solution description in non-technical terms. The non-technical solution defines the requirements for the WbIS under construction. The second step is the transformation of the non-technical solution into a technical solution. The architecture of an instructional system includes the description of four interrelated subsystems:

- **The human subsystem**, which describes the roles for each user involved in the instructional process.

- **The web-based learning resources subsystem**, which are the different online learning resources.

- **The non web-based learning resources subsystem** like text books, DVDs etc.

- **The technical infrastructure subsystem**, which is divided in a common and special infrastructure.
When developing an architecture for a WbIS it is important to keep the entirety as well as the interrelation between the subsystems in mind when breaking up a WbIS [REAV02].

### 3.2.4 LTSA: IEEE P1484-1 Standard

The LTSA describes a high-level architecture and layering for learning technology systems (LTSs). LTSA contains a description of human resources, the infrastructure and learning resources as well as their interactions. The LTSA is divided into 5 layers where each layer is a refinement of the concepts of the above layer. The LTSA describes a solution from an Information Technology perspective and is therefore pedagogically neutral, content neutral, culturally neutral and platform neutral [IEEE02]. Out of the five layers only layer-3 (system components) is mandatory in this standard. This layer identifies the architecture components comprising four processes Delivery, Learning Entity, Evaluation and Coach; and two stores Learning Resources and Learning Records.

Figure 3.2-2 gives an overview of the layer-3 processes and stores.

![Figure 3.2-2: Layer-3 components.](image)

### 3.2.5 Analysis of the WbIS model, UKeU framework and the LTSA

The WbIS model and UKeU framework mainly focus on the area of higher education. In this thesis both the WbIS model and the UKeU framework are used to describe the Domain Model for an MLE that could be used in the educational area as well in the area of corporate businesses. To reach this goal a two step approach is used in this thesis. The first step is to merge the WbIS and the UKeU framework together into one model. In the second step this model will be mapped to LTSA components to demonstrate that this model conforms to IEEE 1484.1. The reason for the latter is because the LTSA is a widely-adopted architecture that makes it easier to build systems that can be integrated with other systems and conforms to the LTSA standard [REAV02].

The WbIS model described by Retalis and Avgeriou gives the technical solutions for instructional problems in the Academic educational area [REAV02] whereas an MLE should provide a technical solution for formal and informal learning in organisations ([IBM06], [Rob05], [Ros06]). To be able to support formal and informal learning the components from the UKeU framework are used as the infrastructure of an MLE.

Figure 3.2-3 shows an MLE model that has merged the WbIS model and the UKeU framework into one model. The layers of the UKeU framework are mapped to the infrastructure subsystem. The portal layer, the common service layer and the database layer of the UKeU framework are mapped to the common infrastructure subsystem and the
learning services of the UkeU framework are mapped to the learning infrastructure subsystem.

Figure 3.2-3: Model for a MLE

Figure 3.2-3 also shows the generic categories for the Learning and Human resources which will be used in the reference architecture.

Table 3.2-1 gives an overview of how the components of the MLE subsystems map to the components in layer 3 of the LTSA. This mapping is used to demonstrate that an MLE conforms to the IEEE 1484.1.

The first column in table 3.2-1 contains the layer-3 components and the first row contains the components from a MLE. The remaining cells show how components from a MLE correspond to layer-3 components of the LTSA.

<table>
<thead>
<tr>
<th>LTSA component</th>
<th>MLE</th>
<th>Technical Infrastructure</th>
<th>Human Resources</th>
<th>Learning Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Entity</td>
<td></td>
<td>Portal, LMS, LCMS, Assessment Tool</td>
<td>Learner</td>
<td></td>
</tr>
<tr>
<td>Behaviour data flow from Learning Entity to Assessment</td>
<td></td>
<td></td>
<td>Learner</td>
<td>Online Material</td>
</tr>
<tr>
<td>Evaluation process</td>
<td></td>
<td>Assessment System</td>
<td>Learner, Instructor, (Line) Manager</td>
<td></td>
</tr>
<tr>
<td>Assessment data flow from Evaluation to Coach</td>
<td></td>
<td>Assessment System</td>
<td>Learner</td>
<td></td>
</tr>
<tr>
<td>Learner information data flow stored and</td>
<td></td>
<td>Administration, LMS, LCMS,</td>
<td>Learner, Instructor</td>
<td></td>
</tr>
<tr>
<td>LTSA component</td>
<td>MLE</td>
<td>Technical Infrastructure</td>
<td>Human Resources</td>
<td>Learning Resources</td>
</tr>
<tr>
<td>----------------</td>
<td>-----</td>
<td>--------------------------</td>
<td>-----------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>retrieved by the Evaluation process</td>
<td></td>
<td>Assessment System, Database(s)</td>
<td>(Line) Manager</td>
<td></td>
</tr>
<tr>
<td>Learner Records database</td>
<td></td>
<td>User Management, Administration, LMS, LCMS, Assessment System, Database(s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learner information dataflow received by the Coach process</td>
<td></td>
<td>LMS, LCMS, Assessment System, Database(s)</td>
<td>Administrator</td>
<td></td>
</tr>
<tr>
<td>Learner information dataflow stored by the Coach process</td>
<td></td>
<td>LMS, LCMS, Assessment System, Database(s)</td>
<td>Administrator</td>
<td></td>
</tr>
<tr>
<td>Learning Parameters dataflow between Learning Entity and Coach</td>
<td></td>
<td>LMS, LCMS</td>
<td>Learner</td>
<td></td>
</tr>
<tr>
<td>Coach Process</td>
<td></td>
<td>LMS, LCMS, Assessment System, Database(s)</td>
<td>Learner, Instructor, (Line)Manager Training Administrator</td>
<td></td>
</tr>
<tr>
<td>Query control flow from Coach to Learning Resources</td>
<td></td>
<td>LCMS, Database(s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learning Resources data store</td>
<td></td>
<td>LCMS, Database(s)</td>
<td>Online and Paper-based material</td>
<td></td>
</tr>
<tr>
<td>Catalogue Information data flow from Learning Resources to Coach</td>
<td></td>
<td>LMS, LCMS</td>
<td>Learner, Instructor, (Line) Manager, Training Administrator</td>
<td></td>
</tr>
<tr>
<td>Locator dataflow sent by Coach process to Delivery process</td>
<td></td>
<td>LMS, LCMS</td>
<td>Online and Paper-based material</td>
<td></td>
</tr>
<tr>
<td>Delivery Process</td>
<td></td>
<td>LMS, LCMS, Assessment System, Portal</td>
<td>Learner, Instructor</td>
<td>Online material</td>
</tr>
<tr>
<td>Locator control flow sent by the Delivery</td>
<td></td>
<td>LCMS, Database(s)</td>
<td>Online and Paper-based material</td>
<td></td>
</tr>
<tr>
<td>Learning Content data flow from Learning Resources to Delivery.</td>
<td></td>
<td>LCMS, Database(s)</td>
<td>Online Material</td>
<td></td>
</tr>
</tbody>
</table>
Table 3.2-1: Mapping between LTSA components and the MLE components presented in this thesis

<table>
<thead>
<tr>
<th>LTSA component</th>
<th>Technical Infrastructure</th>
<th>Human Resources</th>
<th>Learning Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction Context data flow from Delivery to Evaluation</td>
<td>LMS, LCMS, Assessment System Collaboration</td>
<td>Learner, Instructor</td>
<td>Online Material</td>
</tr>
<tr>
<td>Multimedia dataflow from Delivery to Learning Entity</td>
<td>LCMS, Database(s)</td>
<td>Learner</td>
<td>Online Material</td>
</tr>
</tbody>
</table>

The Domain Model for an MLE also serves as the Domain Model for the reference architecture for e-Learning solutions. The Domain Model forms the basis to develop a reference architecture for an e-Learning solution. Section 3.3 gives theoretical underpinnings for a reference architecture. Section 3.4 gives theoretical underpinnings for architectural viewpoints, views and patterns which are important concepts of a reference architecture.

3.3 Documenting a reference architecture for e-Learning solutions

3.3.1 Introduction

Documenting a software architecture means to describe or specify all the architectural elements, diagrams, models decisions, rationale and anything else that concerns the architecture [CBB+05]. Architecture documentation is both prescriptive and descriptive meaning that that for some audience it prescribes what should be true and for other audiences it describes what is true. There are three main uses for architecture documentation ([CBB+05], [BCK03]):

- The architecture documentation serves as a means to introduce and educate users to the system. The users may be new members of the team, external analysts, or new architects.

- The architecture documentation serves as a means of communication between the different stakeholders of the system where a stakeholder is someone who has a vested interest in the architecture. The documentation’s use could vary depending of which stakeholders are involved in the communication. For example:
  - Architects and requirements engineers representing the customer use the documentation for making trade offs among competing requirements.
  - Designers of other systems use the documentation to define the set of operations provided and required for technical compatibility.

- The architecture documentation serves as the bases for system analysis such as the analysis of quality attributes of the system. For example for those interested in checking if the system meets the system quality objectives, the documentation should serve as the input for the architectural evaluation methods.

This section describes the approach I will use for documenting a reference architecture for e-Learning solutions. The approach is based on research in the area of software architecture and reference architectures.
3.3.2 Reference models, reference architectures and software architectures

IEEE mandates that every system has one architecture which is recorded by an architectural description. The architectural description is organised in a set of related views where each view addresses one or more concerns of the system stakeholders [IEEE00]. Figure 3.3-1 gives an overview of the different concepts and their relationship as described in [IEEE00].

Clemens et al. emphasise that documenting an architecture means documenting the relevant Architectural Views and then adding documentation that applies to more than one Architectural View [CBB+05]. The approach from [CBB+05] and [BCK03] is similar to the approach of IEEE. Table 3.3-2 gives an overview of IEEE 1471 requirements and how these are addressed in [CBB+05].

<table>
<thead>
<tr>
<th>IEEE 1471 Requirement</th>
<th>How the IEEE 1471 requirement is addressed in [CBB+05]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stakeholders and their concern</strong></td>
<td></td>
</tr>
<tr>
<td>The standard lists examples that must be addressed for both stakeholders and their concerns.</td>
<td>[CBB+05] captures information about stakeholders and their concern especially how they will use the documentation package for the system which is documented.</td>
</tr>
<tr>
<td><strong>Views</strong></td>
<td></td>
</tr>
<tr>
<td>A View is defined as a representation of the whole system from the perspective of a set of concerns. Each view addresses one or more concerns of the system’s stakeholders.</td>
<td>A view is defined as a representation of a set of coherent architectural elements as written by and read by system stakeholders.</td>
</tr>
<tr>
<td><strong>Viewpoints</strong></td>
<td></td>
</tr>
<tr>
<td>A Viewpoint is a template from which a view can be developed by describing:</td>
<td>[CBB+05] defines View-types where a view-type defines the element types, relationship types used to describe the architecture of a</td>
</tr>
</tbody>
</table>
IEEE 1471 Requirement | How the IEEE 1471 requirement is addressed in [CBB+05]
---|---
• the techniques to create the view; | software system from a particular perspective.
• the way to analyse the view. | Architectural Style: a specialisation of
For each viewpoint the following must be | element and relationship types, together with
specified: | a list of constraints, and how these element
• Stakeholders addressed by the | and relationship types can be used.
viewpoint. | The view-types / architectural styles define
• Concerns addressed by the viewpoint. | the elements / relationships and other
• Language, modelling techniques, | properties that should be used to document a
analytical techniques to be used. | system in accordance to the view-type. Each
• Rationale for selecting the viewpoint. | view-type / architectural style contains a

Table 3.3-2: Overview of IEEE 1471 requirements and how they are addressed by [CBB+05]

According to [CBB+05] view-types are similar to the viewpoints as described in [IEEE00]. The main differences between view-types and viewpoints are that Clements et al. have been documenting view-types that are common in use and have given examples of views. IEEE only provides some examples of viewpoints in [IEEE00]. These viewpoints are only described briefly and no examples are given of views. For the sake of simplicity I will use the term “viewpoints” in the rest of my thesis.

Avgeriou takes the approach from Clements et al. [CBB+05] and Kruchten [Kru95] and proposes a software architecture documentation package that is conformant to IEEE 1471. The software architecture documentation package should contain the following items [Avg03]:

• The system stakeholders and their concerns.
• The definition of Architectural Viewpoints that will address the concerns of the stakeholders.
• The Architectural Views that are specified by the Architectural Viewpoints
• The Architectural Patterns that can be applied in the Architectural Views.
• The quality attributes that should be supported by the architecture.
• A brief description of the implementation constraints.
• Other issues that are of importance to the specific system.

IEEE 1471 and Clements et al. talk about software architectures for systems and not about reference architectures. There are some differences that need to be taken into account if it comes to designing a reference architecture. Rosen describes a reference architecture as a proven template solution for an architecture for a particular domain [Ros06a]. A reference architecture captures the essence of architectures of e-Learning solutions and provides a common vocabulary which can be used to discuss implementations ([Mul07], [Ree02]). An important difference between an architecture for one system and a reference architecture is that the main goal of reference architecture is to provide guidance for the development of architectures of new versions of an e-Learning solution ([Avg03], [Mul07], [Ree02]). Reference Architectures start to appear in organisations when a solution is developed that comprises multiple systems that reside at multiple locations and are developed by multiple vendors [Mul07]. The reference architecture for e-Learning solutions is developed for
multiple customers from multiple organisations and comprise multiple systems that can reside at multiple locations and are developed by multiple vendors.

Figure 3.3-3 shows that relationship between reference models, reference architectures and software architectures [BCK03]. The arrows in figure 3.3-3 indicate that the subsequent concepts contain more design elements.

Figure 3.3-3: Relation between reference models, reference architectures and software architectures

In this thesis a reference architecture for e-Learning solution is developed that addresses the following items:

- A description of the business case that the reference architecture addresses.
- A description of the stakeholders and their concerns that the reference architecture addresses.
- A description of the functional requirements that the reference architecture addresses.
- The definition of Architectural Viewpoints that addresses the concerns of the stakeholders.
- The Architectural Views that are specified by the Architectural Viewpoints
- The Architectural Patterns that can be applied in the Architectural Views.
- The quality attributes that should be supported by the architecture.
- A brief description of the implementation constraints.
- Other issues that are of importance to the specific system.

The main reason for creating a documentation package that addresses these items is that it conforms to IEEE 1471 standard, which is a widely adopted standard [Avg03]. A second reason for creating a documentation package that addresses these items is that it makes it easier to integrate systems that also conform to this standard in an e-Learning solution.

3.3.3 Documentation package for a reference architecture

3.3.3.1 Business case

A business case describes the background and the key drivers that justify the creation of an e-Learning solution. According to the Architecture Forum 5-7 specific key drivers that are critical for success, along with the rational behind the selection should be described by a business case [Mul07]. A business case also describes the benefits an organisation can expect out of the designed solution.

3.3.3.2 Stakeholders and their concerns

Stakeholders and their concerns play a crucial role in the documentation of architectures. As displayed in figure 3.3-1 every system has one or more stakeholders where each stakeholder has one or more concerns relative to that system. Concerns are in this case interests related to the development of the system, its operation or other aspects that are critical for one or more of the stakeholders. Identifying the stakeholders for which the system is going to be
developed should be the first step in designing architectures [Avg03]. The Stakeholders And Their Concerns pattern describes that this group of system stakeholders also serve as the intended audience for the architecture documentation [AGR04]. Avgeriou [Avg03] and the IEEE [IEEE00] have identified the following groups of stakeholders:

- Users of the system, which include the learners, line managers, training coordinators, content authors and system administrators.
- Acquirers of the system, which include the companies that will own the system.
- Developers and maintainers, which include third party software development companies that develop the system on behalf of the system’s business owner and internal software development business units.

Clements et al. have proposed grouping of stakeholders in a more granular way than Avgeriou and the IEEE and include the following groups [CBB+05]:

- Project managers who deal with schedule and resource planning.
- Members of the development team.
- Testers and integrators who deal with testing and integrating black boxes as specified in the architecture.
- Designers of other systems who deal with the integration of systems as specified in the architecture.
- Maintainers of the system who deal with maintenance activities of the system to be developed.
- Customers who are the owners of the system to be developed.
- End users who deal with the system from an operational perspective.

Both end users and customers are more commonly terms used in the literature ([BCK03], [Ros06], [Rob05]). Clements et al. also make a distinction between members of a development team, designers of other systems, application builders and maintainers of a system. In this thesis there is no distinction between a development team, designers of other systems and application builders but these roles are all part of a development team. This thesis focuses on the integration of the Learning Technology components from different suppliers into an e-Learning solution. A team of developers is responsible for designing and building the integration between the different Learning Technology components. The stakeholders and their concerns for e-Learning solutions are described in more detail in section 4.2.

3.3.3.3 Functional requirements

The LTSA [IEEE00] has been established as the basic infrastructure for supporting the technology-based instructional process. Section 3.2.5 describes the relationship between the LTSA and the components used by an e-Learning solution. An e-Learning solution should fulfil certain needs and requirements in a field of increasing demand for effective, fast and pedagogically correct learning experiences. People involved in the decision-making process of creating, managing and administering learning experiences for learners could use an e-Learning solution in the following usage scenarios [APR01]:

- Create, operate and administer on-line learning activities.
- Support collaboration between users.
- Create and deliver questions and tests for learner assessments.
- Organise educational, financial and human resources.
• Administer virtual, distributed learning experiences where learners are geographically scattered and communicate via the Internet.

These diverse usage scenarios correspond to different categories of Learning Technology Systems that could form part of an e-Learning solution. The following categories are identified by Avgeriou et al. ([APR01], [ARS03]):

• General systems which are responsible for creating and managing courses.
• Collaborative learning support systems which provide synchronous and asynchronous tools to users of an e-Learning solution.
• Question and test authoring and management systems which provides tools for the design, construction of quizzes and tests, their on-line delivery, automatic grading and report generation.
• People management systems which deal with resources and financial management.
• Virtual classrooms which establish a virtual space for live interaction for participants in a learning process.

Most Learning Technology Systems that fit one of the above categories support a number of features in order to carry out certain tasks. These features do not discretely belong to particular Learning Technology Systems but can be shared by different categories of Learning Technology Systems. Avgerio et al. and Wilson et al. classify these features in groups such as ([APR01], [ARS03], [WOJP04]):

• Assessment
• Collaboration
• Content management
• Course management
• Group management
• Learner management
• Human resource management
• Packaging
• Resource list and discovery
• Scheduling
• Sequencing

In the UKeU framework these features are grouped together in Learning Technology Systems as described in section 3.2.2. In order to achieve interoperability between the different Learning Technology Systems, which is a major issue in the domain of Learning Technologies, the systems that comprise an e-Learning solution should adopt (parts of) international Learning Technology standards. A single system may implement more than one standard ([Avg03], [Wel07], [WOJP04]).

Table 3.3-3 gives an overview of a mapping between a standard and a Learning Technology Systems.

<table>
<thead>
<tr>
<th>Component</th>
<th>Learning Technology Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>All components</td>
<td>IEEE LTSC LTSA [IEEE00]</td>
</tr>
<tr>
<td>Assessment tools</td>
<td>IMS QTI [IMS06]</td>
</tr>
<tr>
<td>LMS, LCMS</td>
<td>LTSC LOM [IEEE02]</td>
</tr>
<tr>
<td>LCMS</td>
<td>IMS Content Packaging [IMS04], IMS Simple Sequencing [IMS03], ADL SCORM [ADL04]</td>
</tr>
</tbody>
</table>

Table 3.3-3: Mapping between individual category and Learning Technology standard.
The functional requirements that a solution should support are the result of a business requirement analysis done by a business analyst ([Ros06], [Wel07]). The results of a business analysis are captured in a business requirement document that is used by an architect as a starting point for the development of an architecture. Section 4.3.2 describes the functional requirements for the reference architecture.

3.3.3.4 Definition of Architectural Viewpoints

The IEEE 1471 standard recommends the definition of viewpoints that are templates used to define views. The standard does not prescribe a specific set of views but lets the architects of the system free to choose their own views which will suit the stakeholders and their concerns. Clemens et al. also do not require a specific set of views but leave it to the architect to decide which views to use with respect to the stakeholders and their concerns. In this thesis I will use [CBB+05] as a main source for selecting the viewpoints. These viewpoints are described in section 4.4.

3.3.3.5 Architectural Views

After identifying the stakeholders and their concerns, a second very important step is selecting and designing views in such a way that they will address the concerns of the different stakeholders. According to Clemens et al. documenting architectures is all about documenting views and adding documentation that applies to more than one view [CBB+05]. Viewpoints, views and patterns are further explored in section 3.4. Clemens et al. provide some advice on how to reconcile their view-types and architectural styles and the architectural views from Kruchten. The reconciliation between the Architectural Views with the view-types and Architectural Styles for the reference architecture of e-Learning solutions is discussed in more detail in section 3.4.3. As all views describe system elements and their relationships an architect can use a common organisational scheme to document them. According to Clements et al. the organisational scheme should contain the following [CBB+05]:

- Primary presentation. The primary presentation should show all the elements and their relationships. The primary presentation is most of the time graphical and should contain an explanation of the notation used in the presentation.
- Supporting documentation. This documentation should explain and elaborate the information in the primary presentation.

3.3.3.6 Architectural Patterns

Architectural Patterns should have a special place in the documentation because they provide solutions to problems on the architectural level. Each of the patterns should be appropriate for a specific view and can be used to organise the architectural elements in the view. The architectural patterns used in each view give the stakeholders a common vocabulary so that they can speak the same language when discussing various aspects of the system [AGR04].

Avgeriou and Zdun have used have used “classical” Architectural Patterns from Buschmann et al. [BMR+96], Clements et al. [CBB+05] and others because they are well established in the software architecture community [AVZD05].

3.3.3.7 Quality attributes

Most of the architectural design tends to focus on the functional requirements of a system and not as much on the non functional requirements of a system. Functional requirements, as described in section 3.3.3.3, describe what the system has to do and non functional
requirements describe the tolerances, boundaries and standards that must be adhered by the system delivering its functionality to its users. Non functional requirements are often referred to as the quality attributes or the qualities of a system. Achieving the quality attributes is something which needs to be taken into consideration throughout the design, implementation and deployment of the system. Architectures are critical to the realization of the quality attributes of interest in a system. Architectures by themselves can not achieve the qualities; they provide the foundation of the qualities of a system [BCK03]. Dyson and Longshaw describe non functional characteristics for internet based technology systems in [DYLO04] and Avgeriou and Skordalakis describe non functional characteristics for LMSs in [ARS02]. In this thesis the focus is on quality attributes for internet based technology systems because e-Learning solutions are examples solutions that are based on internet technology. The quality attributes are discussed in more detail in section 4.3.3

3.3.3.8 Implementation constraints

In many cases an architect will have a certain implementation strategy in mind for a system or may know constraints that the implementation of a system must follow [CBB+05]. Implementation decisions determine the development of a system and should therefore be taken during the architectural design phase and not in a later phase of a system [Avg03]. Avgeriou has chosen to use the component-based paradigm using object-oriented techniques like UML and Java as programming language in [Avg03]. These decisions highly determine the development of the system in later phases of the development cycle. Compared with an architecture of a system a reference architecture is a software architecture at a higher level of abstraction that does not contain these kind of implementation details ([Avg03], [BCK03]). The implementation constraints of reference architecture for e-Learning solutions are therefore more generic but the two business cases used in this thesis to evaluate the reference architecture will have concrete defined implementation decisions. These specific implementation decisions are described in section 9 and 10. The implementation constraints for the reference architecture are described in more detail in section 4.3.4.

3.3.3.9 Other issues

This section is used by Avgeriou to address issues, not described in one of the other sections that are important for the system under development [Avg03]. In the case of the architecture for an LMS issues like the adoption and implementation of international Learning Technology standards such as IEEE LTSA, ADL SCORM and IMS QTI are important [Avg03]. The business cases described in section 9 and 10 have a section that addresses the adoption and implementation of the required standards. Clements et al. suggest that this part of the documentation package could be used for adding a glossary of terms and a list of acronyms [CBB+05]. The documentation package of this thesis has a glossary of terms and a list of acronyms in appendix A.

3.3.3.10 Evaluation and conformance criteria for the reference architecture

This section describes the evaluation criteria that are used to assess the reference architecture. Architectural assessment is an activity of an architecting process to evaluate the degree of fulfilment of the quality, or non functional requirements. The difficulty with the assessment of quality attributes is that they cannot be measured after a solution is built because it is too expensive to fix a solution when it is completed. Quality attributes are more abstract design decisions in the mind of a solution architect. To be able to assess quality attributes of an architecture special assessment techniques are created for the
purpose of evaluating these attributes ([ARS02], [Bos00], [DRos06]). In this thesis the
reference architecture is evaluated by using two business cases instead of building an
architectural prototype of the reference architecture. Both business cases are examples of e-
Learning solutions built by British Telecom (BT). A second approach that is used is based
on an experience based assessment of the quality attributes of the architecture. In this
approach the quality attributes are assessed based on the experience of architects and can be
found in the literature ([ARS02], [BCK03], [DYLO04]).
The reference architecture should also conform to LTSA and IEEE 1471-2000 for
architectural description because this makes it easier to integrate systems that also conform
to these standards ([ARS03], [IEEE00]).

3.4 Architectural Viewpoints, Views and Patterns

3.4.1 Introduction
This section discusses different aspects of viewpoints, architectural views and architectural
patterns. These concepts are the concepts of documenting architectures ([Kru95],
[AVZD04], [CBB+05], [BCK04]).

3.4.2 Architectural Viewpoints
Viewpoints are a means to focus on certain aspects of the architecture and are determined
by the concerns of the stakeholders. The viewpoints can therefore be used in
communicating these aspects with stakeholders. Views are specified by viewpoints that
prescribe the elements and their relationships that are provided by the view. In this sense a
view conforms to a viewpoint [IEEE00].
Avgeriou has defined five viewpoints: use-case viewpoint, logical viewpoint, deployment
viewpoint, implementation viewpoint and the data viewpoint [Avg03]. These viewpoints
relate to the 4+1 view approach of Kruchten and the Rational Unified Process (RUP)
[Kru95, Kru04]. Clements et al. have defined three different view-types: the module view-
type, component and connector view-type and the allocation view-type [CBB+05].
Clements et al. do not prescribe a set of views as Avgeriou in [Avg03] but recommend
choosing a set of views that will address the concerns of the stakeholders of the architecture
under development [CBB+05]. In this thesis the approach of Clements et al. is used because
it gives more flexibility to add new viewpoints that can be used in a reference architecture.

3.4.3 Architectural Views
One of the, perhaps most important, concepts related to documenting architectures is views.
According to [CBB+05] documenting an architecture is a matter of documenting the right
views and adding information that relates to more than one view. In ([CBB+05], [BCK03])
a view is defined as the representation of a set of elements and their relationship. A view
documents a particular aspect of a system’s architecture while, intentionally suppressing
other aspects. It is generally accepted that a system’s architecture is usually too complex to
be described in a simple one dimensional fashion but must be described in multiple views.
Views can be categorised according to three dimensions (also called view-types)
([CBB+05], [BCK03], [BOHO06]):
- Module category with views about the structure of the implementation.
- Component and connector category with views about the run-time units or run-time
  behaviour and interaction.
• Allocation category with views about how the software relates to its deployment and execution environment.

Kruchten has described four main views for documenting a software architecture and a fifth view that ties the other four together [Kru95]:

• Logical view: The logical view supports the behavioural requirements, the services the system should provide to the users.

• Process view: The process view addresses some of the non functional requirements such as: performance and availability and it addresses issues such as concurrency and distribution, system integrity, fault tolerance and how the main abstractions from this view map to the processes within the process architecture.

• Development view: The development view focuses on the organisation of the modules in the software development environment. The software is divided into smaller chunks which can be developed by developers. These subsystems are organised in a hierarchy of different layers.

• Physical view: The physical view presents the non functional requirements such as availability, reliability, performance and scalability.

The fifth view is the use case view that shows that the elements of the 4 views can work together by the use of a set of important use cases (scenarios). Avgeriou uses these 4+1 views for documenting the reference architecture for an LMS in the case study used in [Avg03, AGR04].

The views defined by Clements et al. [CBB+05] are far more fine-grained than the views defined by Kruchten in [Kru95]. Another difference between the views of Clements et al. and Kruchten is that Kruchten prescribes 5 views whereas Clements et al. do not prescribe a set of views.

Views of large or complex solutions could contain a large amount of elements. Showing these elements in a single presentation, along with their relationship, could end up in an overkill of information for stakeholders who only have concerns with a certain part of the solution. Views are therefore presented in smaller digestible “chunks”. These chunks are called view-packets. The documentation of views or view-packets can be placed into a standard form containing the following parts [CBB+05]:

• Primary organisation. The primary organisation gives a graphical or textual overview of the elements and relationship used in a view-packet.

• Element catalogue. The element catalogue details the elements described in the primary presentation. The element catalogue should also include the elements that are important to the view-packet but were not mentioned in the primary presentation.

• Context diagram. The context diagram shows how parts of the solution in the view-packet relate to its environment.

• Architectural background. The architectural background explains why the design is as it is.

3.4.4 Patterns, pattern stories and a pattern language for e-Learning solutions

Another important concept associated with the documentation of software architectures are architectural patterns. The concept of architectural patterns is applied in the literature in two different ways: architectural patterns and architectural styles. Both architectural patterns as well as architectural styles refer to recurring solutions that solve problems at the
architectural design level and they also provide a common vocabulary to facilitate communication [AVZD05]. In this thesis architectural patterns and architectural styles are treated as the same concepts and the term architectural pattern is used as proposed by [AVZD05]. Furthermore, patterns are considered to be architectural they refers to a problem at the architectural level and covers the overall system and not only some individual subsystems [AVZD05].

A pattern should teach in the sense that it should be clear enough for readers to understand the context, the problem, the forces and the solution in such a way that readers can adapt the problem and solution description to meet their own needs [Ris98]. The name of a pattern should be self explanatory and help relate the pattern to other patterns in such a way that it can be used in pattern sequences, pattern stories or in pattern languages [BHS07b].

Almost all the approaches to describe patterns have used the so-called Alexandrian format for their pattern templates [Ale79]. Each pattern has a three-part scheme, a context; a problem and a solution where each of the three parts can be divided in different aspects. In this thesis the focus is on re-using architectural patterns that are commonly known and accepted by the pattern community. Therefore only a short description is provided of each of the patterns used for documenting the reference architecture. The following fields are used to describe the patterns:

- **Pattern name**: Name of the pattern;
- **Context**: Description of the context of the problem;
- **Problem**: A statement of the problem that this pattern resolves;
- **Solution**: Description of the solution of the problem;

These fields are commonly used by the authors in the architectural pattern domain ([BHS07a], [BMR+96], [DYLO04], [MAE02], [MED+06], [HOW004]).

Patterns can be connected through a pattern story using patterns from a pattern collection. A pattern story is a story about system development in which design questions are asked and answered and structures are assembled for specific purposes [BHS07b]. Pattern stories are like diaries that tell their readers how patterns guide the development of a system or subsystem. Pattern stories also discuss what problems were resolved and in what order during the construction of the system. Pattern stories are tightly coupled to the context, requirements and the development constraints of a specific system. Two different systems from the same domain, but with a different context, requirement and development constraints, may follow a different pattern story simply because the differences suggest the use of different patterns or a different pattern ordering. In this thesis pattern stories are used to document the views of the architecture of both business cases.

Pattern languages describe how a collection of patterns are connected to build systems or solutions in a specific domain. Each pattern language serves as an overview, introduction to and communication vehicle for the best practices of its respective domain. The domain characterises the problems that can arise and their solutions, the independencies between the problems and solutions and in which order problems could be resolved. The language ensures that the problem is not resolved in isolation but under the consideration of its surrounding problems and their solutions [BHS07b].

In this thesis architectural patterns are used to design the relevant parts of the reference architecture. The reason for using architectural patterns is that they provide proven solutions for problems in a specific context [Mul07]. This thesis also investigates if it possible to
propose a pattern language that can be used to design a reference architecture for e-Learning solutions.
4 Reference Architecture for e-Learning solutions

4.1 Business case for a reference architecture for e-Learning solutions

4.1.1 Introduction
E-Learning puts the control where it’s needed, in the hands of end users (learners), so that they can learn wherever and whenever it suits them best. It also gives new resources to instructors such as interactive multimedia, simulations and other emerging learning techniques. Furthermore, e-Learning solutions provide managers with the capability to monitor the progress of learners, assess course effectiveness, identify skills gaps on an organisational level and create development plans on an organisational and personal level to address the skills gaps ([Rob05], [Ros06]). This section describes a business case for a reference architecture for e-Learning solutions.

4.1.2 Domain Model for an e-Learning solution
An e-Learning solution in this thesis is defined as a solution in which a Managed Learning Environment (MLE) is developed (section 2.1). The Domain Model ([Fow03], [BHS07a]) of an MLE that is proposed in this thesis is based on the UkeU framework and SUN’s e-Learning framework (section 3.2.2), the Model for Web-based Instructional Systems (section 3.2.3) and the LTSA Standard (section 3.2.4). A model for an MLE was created in section 3.2.5 in which an MLE is comprised of three subsystems: the infrastructure subsystem, the learning resource subsystem and the human resource subsystem. The human resource subsystem describes the users that interact with components of the solution or are responsible for the development and maintenance of the solution. The learning resource subsystem is a data store for all the on-line and paper based courseware material used by the learners of the solution for their learning. The infrastructure subsystem is based on the UKeU framework and consists of learning services and common services.

4.1.3 Rationale for creating a reference architecture
The BT LS team develops e-Learning solutions that provide customers with these functions. So far the team has developed new solutions without re-using what has been implemented and learned in previous solutions. This is an inefficient way of developing new solutions in which development work is done over and over again without re-using what was done in the past. A reference architecture, as described in section 3.3.2, addresses these issues by re-using architectural patterns from different domains to design the views of the reference architecture.

4.1.4 Benefits of a reference architecture for e-Learning solutions
The benefits of e-Learning solutions that are developed by using a reference architecture can be found into two areas. The first area describes the business benefits of an e-Learning solution developed by BT LS for a customer and the second area describes the benefits from an architectural point of view of using a reference architecture to design architectures for new e-Learning solutions.

Business benefits of an e-Learning solution
The implementation of an e-Learning solution developed by BT LS could have the following benefits for a customer [BTLS05]:

- The solution helps to cut real training costs.
The solution can be used by organisations to prove compliance with industry standards.

The solution can be fully customised to meet the training needs of the customers’ business.

The solution offers a rich learning environment for the end-users.

The solution is built using best-of-breed learning technology applications of suppliers which will be around for the long-term.

The solution provides comprehensive management reports.

The solution can be fully integrated with corporate systems.

Benefits of a reference architecture for e-Learning solutions

A reference architecture for e-Learning solutions has the following benefits for the BT LS team and their customers (see section 2.2):

- The BT LS team can provide better return on investment for its customer.
- The BT LS team can enable faster deployment of technology for its customer.
- The BT LS team can provide a modular flexible technology base for its customers.

If we draw a conclusion from these benefits we could say that a by using a reference architecture for e-Learning solutions a customer has the benefits of an e-Learning solution but also the benefits that BT LS can provide; a better return of investment, a faster deployment of technology and a flexible technology base.

4.1.5 Scope of the reference architecture

The focus of the reference architecture described in this chapter is on the integration of the different learning technology applications with one another and with corporate applications of an organisation. The reason to focus on the integration of applications is because BT LS builds solutions using commercial off the shelf components (COTS) or open source applications. A complete description of a business model for e-Learning solutions is out of scope of this reference architecture.

4.2 Stakeholders and their concerns

4.2.1 Introduction

Stakeholders and their concerns play a crucial role in the documentation of architectures as described in section 3.3.2.2. Concerns are in the interests of the stakeholders related to the development of the system, its operation or other aspects that are critical for one of more of the stakeholders.

This section describes the stakeholders and their concerns for e-Learning solutions. Section 4.2.2 identifies the stakeholders of e-Learning solutions and section 4.2.3 describes the concerns of these stakeholders.

4.2.2 Stakeholders

The stakeholders that are considered in this thesis are:

- The Customer / Business owner of the solution;
- End user of the solution;
- A development team responsible for building an integrated e-Learning solution;
- A maintenance and support team responsible for the maintenance and day to day support of the system.
The selection of these stakeholders is based on the result of a mining exercise of four business cases of BT LS of which case A and case B are examples. The selection of stakeholders is similar to the stakeholders that are identified by Weller [Wel07]. The next section will describe the concerns of the stakeholders.

4.2.3 Concerns
Each stakeholder has certain concerns regarding the e-Learning solution being developed, these concerns relate to for instance the cost of the solution, the performance or the way the solution is build. This section gives an overview of the concerns of the stakeholders of an e-Learning solution.

Customer and business owner of the solution
The customers and business owners of the solution are interested in the general idea of the solution, the cost of the solution, the return of investment (ROI) of the solution, what they are getting for their investment and the time and effort it will take to develop and implement the solution. Customers and business owners are interested in the reporting capabilities of the solution so that they can measure the effectiveness of what is learned. Customers and business owners also need assurance that the solution runs using the corporate infrastructure and integrates with for instance with their HR system ([BCK03], [Avg03], [Ros06]). Concerns that the customer and business owner would like to see addressed therefore are:

- What is the general idea behind the solution?
- What are the costs of the solution?
- How long will it take to deploy the solution?
- What are the reporting capabilities of the solution?
- Are the functional requirements implemented in the solution?

End users
The end users of an e-Learning solution could have different roles in an e-Learning solution. In a typical e-Learning solution the following roles can be identified ([SABAa], [SUMTa]):

- Learner
- Instructor
- (Line) Manager
- Training Administrator
- Administrator
- Author

The end-users are interested in the functionality such as the ability to register learners to a course, the ability to create new courses, the ability to report on the progress of direct reports or the ability to launch a course. End-users would like to have a seamless experience if it comes to the usage of the different components of the solution. Concerns that the end-users would like to see addressed are ([IBMa], [Ros06], [Rob05]):

- What is the general idea behind the solution?
- Does the solution provide end users relevant learning, when they need it in order to perform their jobs well?
- Does the solution have capabilities to link learning to skills and jobs or functions?
- Does the solution have reporting capabilities that can measure end users’ progress on their personal development plan and the effectiveness of learning programs?
- Is the solution easy to use; is it easy for end users to find their way through the different parts of the solution?
Development team
Members of a development team could have different roles in the development of an e-Learning solution. The following roles could be identified [CBB+05]:

- Solution architect
- Solution designer
- Developer

A member of a development team could be responsible for the design of a new interface for an existing component of the solution or for the integration of a new component such as a commercial off-the-shelf component. For a development team member it is therefore important to understand which components are used, how these components relate to each other, which parts can be re-used or a description of the interfaces of existing or new components. Team members of a development team will want to know ([CBB+05], [BCK03], [Avg03]):

- What is the general idea behind the solution?
- What components comprise the solution?
- What are the non-functional requirements that the solution must support?
- In which components is the solution decomposed?
- What is the relationship between the components of the solution?
- How is the functionality allocated to the different components of the solution?
- How are the different components layered across the solution?
- Which integration styles are used in the solution?
- How do the independent components in the solution interact with each other?
- How are individual components decoupled from each other in the solution?
- What data is stored by which components in the solution?
- How is the data created, accessed and updated by the different components in the solution?

Maintenance and support team
A member of the maintenance and support team is responsible for the maintenance and support of the different components that comprise the solution. A member of the maintenance and support team is interested in the services the solution provides to the customer and end-users, the components that comprise the solution or the way the components communicate or interact with another. The maintenance and support team will want to know ([CBB+05], [BCK03], [Avg03]):

- What is the general idea behind the solution?
- What components comprise the solution?
- In which components is the solution decomposed?
- Which integration styles are used in the solution?
- How do the independent components in the solution interact with each other?
- What network protocols are used by the different nodes on the network?
- Which component is installed on which node on the network?
- What data is stored by which components in the solution?

Summary of the concerns addressed by the architecture
In this thesis I will address the following concerns:

1. What is the general idea behind the solution?
2. What are the costs of the solution?
3. How long will it take to deploy the solution?
4. What are the reporting capabilities of the solution?
5. Are the functional requirements implemented in the solution?
6. Does the solution provide end users relevant learning, when they need it in order to perform their jobs well?
7. Does the solution have capabilities to link learning to skills and jobs or functions?
8. Does the solution have reporting capabilities that can measure end users’ progress on their personal development plan?
9. Is the solution easy to use; is it easy for end users to find their way through the different parts of the solution?
10. What components comprise the solution?
11. What are the non-functional requirements that the solution must support?
12. In which components is the solution decomposed?
13. What is the relationship between the components of the solution?
14. How is the functionality allocated to the different components of the solution?
15. How are the different components layered across the solution?
16. Which integration styles are used in the solution?
17. How do the independent components in the solution interact with each other?
18. How are individual components decoupled from each other in the solution?
19. What data is stored by which components in the solution?
20. How is the data created, accessed and updated by the different components in the solution?
21. What network protocols are used by the different nodes on the network?
22. Which component is installed on which node on the network?

4.3 Functional requirements, quality attributes and implementation constraints

4.3.1 Introduction
A lot of e-Learning solutions mainly focus on satisfying functional requirements, such as course scheduling, creation and packaging of online course material, communication and collaboration between various users etc. Satisfying quality attributes (non-functional requirements) are usually overlooked and underestimated. The main reason for this is that quality is seen as the prime interest of software vendors who usually focus on functionality because it is more tangible and a better argument for marketing purposes. This could result in inefficient systems, poor software and poor business quality ([BCK03], [ARS02]). Therefore non-functional requirements such as availability, performance, security etc should be described and agreed upfront before an architect starts with the design of an architecture. This section describes the functional and non-functional requirements that should be supported by an architecture of e-Learning solutions. It also describes the implementation constraints of the reference architecture and how the reference architecture could be instantiated.

4.3.2 Functional requirements
An e-Learning solution needs to be able to support and manage a blended approach in its learning activities that incorporates all types of learning, including traditional classroom training, online course delivery and virtual classroom training. To achieve this a broad range of functions is required. This section describes the functional requirements that should be fulfilled by an e-Learning solution. In this thesis the focus is on the components of the learning infrastructure of an MLE because it is envisaged by BT LS that an MLE is
always integrated with the components of the corporate infrastructure such as a portal, Human Resource (HR) system or a Mail System.
The functional requirements are grouped around the areas that should be addressed by an e-Learning solution.

Management and administration
The management component of an e-Learning solution comprises four main functions that are linked together: learning management, skills management, performance management and user profile management. The management component should be tightly integrated with the corporate HR-system and provide the information for e-Learning analysis and reporting. The following requirements are identified for each of the functions ([Ros06], [Rob05], [VOC+06], [Wel07]):

Learning management

- The learning management module should provide facilities to analyse training needs, course and programme definition and management, resource management and learner profile management.
- The learning management module should be integrated with an end user’s calendar and include routing and notification facilities to ensure users receive reminders to complete a course by a certain date or to attend a class on a certain date etc.
- The learning management module needs to provide security at both user and content level. This should ensure that users only have access to information to which they are entitled and that they can alter data or content within the system based on their privileges.
- The learning management module should hold a user profile for each end user that holds information about their skills, competencies, learning plans and learning history. This should be tightly integrated with the corporate HR-system and LDAP directory.

Skills Management

- The skills management module should provide the facilities to set-up a competency framework and map these competencies against every person in the organisation.
- The skills management module should be tightly integrated with the learning management module to make it possible for end users to apply for training that will fill in gaps identified by a skills gap analysis.

Performance management

- The performance management module should provide the facilities to perform a skills gap analysis to identify the areas within an organisation where there are weaknesses or dependencies on too few employees.
- The performance management module should be tightly integrated with the skills management and learning management modules.
- The performance management function should provide the facilities to create and maintain personal development plans for each of the employees based on an end user’s performance appraisal, business objectives and personal objectives.

Analysis and reporting

- The analysis and reporting module should provide extensive tracking of all activities within the system.
- The analysis and reporting module should provide standard reports and options to create custom made reports or options to integrate with corporate reporting capabilities.
Content Management
- The content management module should provide functions to handle and manage course content throughout its lifecycle, from creation through publication and revision to archive.
- The content created by content authors should be compliant with learning technology standards such as AICC and SCORM.
- The content management module should be able to manage several types of content including third party content, assessment content and recorded content (recorded virtual classroom sessions).
- The content management module should be able to manage courses as a collection of re-usable objects (RLO) in order to personalise courses to the individual needs of the learners.

Collaboration
- The collaboration module should provide capabilities for synchronous (virtual classrooms, video conferencing) and asynchronous (wikis, blogs, e-mail, discussion threads) collaboration of end users.

Assessment
- The assessment module should provide the facilities to create, deliver and manage assessments and record the result for a learner.
- The assessment module should provide the facilities for managers to monitor the progress and performance of their direct reports.

Integration
- The solution should have the capability to integrate with the corporate HR-system.
- The solution should have the capabilities to integrate with the corporate solution for single-sign on.
- The solution should have the capabilities to integrate with the corporate e-mail and calendaring applications.

4.3.3 Non-functional requirements
Functional and non-functional requirements or quality attributes of a reference architecture will have an impact on one another. Also, no non-functional requirement or quality attributes can be maximised in a system without sacrificing some other quality or qualities. There is always a trade-off while choosing the different non-functional requirements ([BCK03], [DYLO4], [ARS02]). A whole range of quality attributes are proposed by various researchers, as well as international standards. Fortunately, these sets of non-functional requirements relate to the same concepts [ARS02].
Avgeriou et al. categorise quality attributes into three classes: quality attributes that are important at run-time, development qualities and business qualities [ARS02]. Bass et al categorise qualities into system qualities and business qualities such as cost and time to market. Dyson and Longshaw have grouped non-functional requirement into the following three classes:
- system performance: availability, performance, short-term scalability
- system control: security, maintainability, manageability
- system evolution: flexibility, portability, long-term scalability
In this thesis the classification from Dyson and Longshaw is used because e-Learning solutions are all web-based, high capability solutions. The focus in this thesis is on the core
non-functional requirements as described by Dyson and Longshaw with the addition of cost as business quality. The reason for the latter is because cost is a non-functional requirement that could have an impact on all other non-functional requirements [DYLO04]. In this schema I replaced portability by interoperability and I replaced manageability by usability. Portability is replaced by interoperability because the integration aspect of the different components in an e-Learning solution is very important. Further more portability is less of an issue for web based systems. Manageability is replaced by usability because in this thesis the focus is not on monitoring a system or altering a system’s runtime behaviour. Usability is important because the ease of use of an e-Learning solution will have a high impact on the acceptance of the solution by end-users of the solution. This subsection describes the different non-functional requirements that the reference architecture should support ([ARS02], [BCK03], [DYLO04]).

**Availability**

An e-Learning solution must be robust enough to serve the diverse needs of thousands of learners, administrators, managers, content builders and instructors simultaneously. Availability is an indication of the “up-time” of the components in an e-Learning solution. The availability is measured by the length of time between failures. Availability of an e-Learning solution can be enhanced by duplicating critical components to the solution. The availability is based on the reliability of the individual components and the robustness built into the architecture ([BCK03], [DYLO04], [ARS02]).

For end-users of an e-Learning solution it is important that the components that comprise the solution are available and the down time of the different components is minimal. For instance every time an LMS is unavailable a learner cannot launch an on-line course, an administrator cannot enrol learners onto courses etc. Another important issue for end-users is the fact that the failure of one component does not affect the other components. For instance when the LCMS fails to work it should not stop the LMS or Assessment System from working.

The availability of an e-Learning solution for a customer is normally described in a Service Level Agreement (SLA) between a customer and BT LS. In an SLA BT LS and a customer agree on time windows in which a system can be taken down for maintenance and upgrade activities. Outside this agreed down time window the solution should be available for at least 99.1% of the time [DYLO04].

**Performance**

The performance of an e-Learning solution is the ability of the components that comprise the solution to provide a response in a timely manner. A requirement for the performance of a reference architecture for e-Learning solutions is that it is expected that all users of the solution consistently enjoy good performance. It is important when designing the reference architecture to keep the overall throughput of the different components in the solutions in balance with perceived performance by the users. It could be quite possible to create a solution where every component serves only a few users. This would give a well perceived performance but poor throughput ([BCK03], [DYLO04], [ARS02]).

The response of a component of the reference architecture can be characterised by [DYLO04]:

- Latency: time between the arrival of a request and the response of the component.
- Throughput: the number of transactions a component can do in a second.
- The number of events not processed because the component is too busy.
- The data that gets lost because the component was too busy.
When learners, instructors, managers or courseware authors make a request to a component of an e-Learning solution they expect a response from the component within a reasonable amount of time. For instance when a learner searches for a learning activity he expects a search result within seconds or when a manager wants a report of the performance of his direct reports he does not expect that it will take an reporting tool minutes to produce the report.

For a typical e-Learning solution the reference architecture should be designed in such a way that it can cope with 1% - 2% of concurrent users without any performance loss. Concurrent users are users performing the same interaction with a system at the same time i.e. 2% of the registered users are launching a course at the same time.

Usability
Usability is concerned with how easy it is for an end-user of an e-Learning solution to learn to use a component’s interface to perform task and the support a component provide. To be able to assess the usability of this reference architecture it should be broken into the following areas ([ARS02], [BCK03]):

- How easy is it for an end-user to learn the user’s interface?
- How fast does a component respond to end-user’s request?
- Can end-users remember the use of a component between uses of the component?
- Does a component anticipate in preventing common end-user’s errors?
- Does the solution make an end-user’s job easy?

These questions should be asked and answered when assessing the usability of reference architecture.

Usability is especially important for the end-users of an e-Learning solution as they are responsible performing day to day tasks using the solution. Learners are interested in how easy it is to register themselves onto courses, or the easiness to participate in online courses. A manager is interested in how quick he can get up-to-date information on the performance of his direct reports.

Security
An e-Learning solution should not allow unauthorised usage of online content, resources and back-end functions that comprise the solution but the solution should also provide its services to legitimate users ([ARS02], [BCK03], [Wei05]). Security that should be supported by the reference architecture can be characterised by the following properties of components ([BCK03], [Wei05]):

- Non-repudiation is the property that a transaction cannot be denied by any the parties involved in the transaction.
- Confidentiality is the property that the data or services are protected against unauthorised access.
- Integrity is the property that the data or services are delivered as intended.
- Assurance is the property that the parties involved in a transaction are who they say they are.
- Availability is the property that the component will be available for legitimate use.
- Auditing is the property that a component tracks enough information to reconstruct activities.

Customers or business owners require that an e-Learning solution has security measures in place that only gives users access to those components that they are allowed to use, that users only have access to data that they are allowed to access and that users only have access to courses that they are supposed to launch. End-users require that their privacy is
protected by the solution and that they have access to the functionality and courses they want to use.

**Scalability**
The infrastructure of an e-Learning solution should be able to expand or scale to meet future growth, both in terms of the volume of courses and the number of the learners. Whereas performance is concerned with ensuring that a predicted group of users get a good response time, while using components of the e-learning solution, scalability is concerned with ensuring that the performance does not degrade as the number of users grows. The scalability quality requirement of the reference architecture should deal with short term growth (the number of users using the components over the course of a week, month or year) and long term growth (the number of users that increases as a result of the growing use of services) [DYLO04].

End-users are less interested in the scalability of an e-Learning solution as long as the learning technology components that comprise the solution are available and they are satisfied with the overall performance. Maintainers of the solution are interested in the scalability of the solution to meet the expectation of the end-users regarding the performance. Maintainers need to know which options the solution supports to meet the short and long term scalability requirements.

**Flexibility**
An e-Learning solution is not developed and deployed to be left unchanged for the duration of its entire operational lifetime. Flexibility is concerned with the ability to change the functionality of an e-Learning solution because of changing requirements of the customer. Changing functionality of the solution means that it should be possible to add, remove or change components to the solution without being forced to rebuild the complete architecture of the solution ([ARS02], [BCK03]).

The customer or business owner of an e-Learning solution is interested in the easiness of adding new functionality to the solution when the requirements change. For instance when a customer wants to be able extend an e-Learning solution with the option to setup virtual classrooms he expects that the LMS can be integrated with a virtual classroom application. Developers are interested in the options to add new functionality to the solution either by developing a new learning technology component or a new interface to an off-the-shelf component.

**Interoperability**
Standardisation plays an important role in enterprise wide solutions such as e-Learning solutions in which many vendors are involved. Achieving enterprise wide interoperability requires industry standardisation of formats, protocols, services and the ability to exchange data and request functionality. For enterprise software this means interchangeability and componentisation. By standardisation of formats, protocols and services individual interfaces can be replaced by other components. This isolates the effect of failures, permits each component to be maintained and updated independently and reduces the dependence on a single vendor. The ability of components to exchange data is governed by the specification of component interfaces and their interactions. The impact of this requirement on the components in an e-Learning solution, such as learning management systems, employee information systems, course management systems, knowledge management
systems and content management systems, is that an interface should be specified in such a way that these components can exchange data ([ARS02], [BCK02]). Developers are interested in the standards that learning technology components support to be able to integrate the different components in one solution. For instance developers would like to know if learning technology components support web-services to exchange information with each other. Courseware authors also want that the solution supports learning technology standards such as SCORM.

**Maintainability**
Components of an e-Learning solution will always need upgrading and maintenance fixes during the live-cycle of the solution. Maintainability is the ability to upgrade and apply fixes quickly and cost-effectively. Flexibility and maintainability are somewhat related to one another. Flexibility implies that something is changed and measures how well developers or the maintenance team can re-configure the (parts of) solution in face of the change. Maintainability is more about how easy or hard it is to fix problems in an existing solution or upgrade (parts of) a solution of to apply patches to components of a solution ([ARS02], [DYLO04]). Suppliers of learning technology components have a road map with which they provide customers future upgrades of their components and the enhancements the upgrades will provide to the customers. Fixes or patches will also be provided to customers when defects are found in their components. For the maintenance team it is very important to know how easy or difficult it is to upgrade a component or apply a fix for a defect. How easy the latter can be done depends on the way the component are integrated and how heavily the components are customised to meet the customers’ requirements.

**4.3.4 Implementation constraints**
There are several implementation constraints that should be taken into account by an architect when designing an architecture for an e-Learning solution. The implementation constraints for this reference architecture are described in a generic way and are refined by the architectures of instances of this reference architecture. The following implementation constraints should be taken into account for the design of the reference architecture:

- An architecture for an e-Learning solution never stands on its own but should fit within the enterprise architecture of a customer organisation.
- The design of the reference architecture is not based on particular technologies such as the .NET of Java technologies. The choice should be based on what is supported by the enterprise architecture of a customer organisation.
- The design of the reference architecture should be based on the usage of commercial off the shelf components or open source components.
- The components that comprise an MLE should be integrated with one another and with components of enterprise architecture using one or more of the approaches as described in section 3.1. The choice for a specific approach should be based on what is supported by the enterprise architecture of a customer organisation (functional requirement for integration is described section 4.3.2).
- The components that comprise the MLE should adopt the learning technology standards as described in section 3.3.3.3.
- The components that comprise an MLE should ideally run on the core operating systems (Windows, Linux or Unix) and using a core database systems (MS SQL or Oracle).
• All components that comprise an MLE should be accessible by the web to achieve the objective that an end user should be able to access the solution from any place.
• An MLE should be designed following the accessibility guidelines from section 508 in the US (www.section508.gov) and W3CAA (www.w3.org/WAI) in Europe. The components of an MLE should allow assistive technology such as screen readers and speech recognition packages to work effectively.

4.3.5 Architecture instance for an e-Learning solution
This subsection describes the steps that should be used to create an instance of the reference architecture for e-Learning solutions.
1. Describe a business case and a scope that adopts the Domain Model (see section 4.1) as underlying model for this new architecture instance.
2. Describe the stakeholders and their concerns for this new architecture instance (see section 4.2).
3. Describe the implementation constraints for this new architecture instance (see section 4.3.4).
4. Describe the functional and non functional requirements that should be used for the design of this new architecture instance (see section 4.3.2 and 4.3.3).
5. Design the views for this new architecture instance (see section 4.5).
6. Evaluate the new architecture instance (see section 4.6).
7. Describe the evolution of this new architecture instance.

4.4 Architectural viewpoints

4.4.1 Introduction
In this thesis the view-types of Clements et al., as described in section 3.4.2, are used because they provide the flexibility to choose a set of views that address the concerns of the stakeholders as identified in section 4.2. For the sake of simplicity the term viewpoints is used when referring to view-types because there is a clear similarity between the two terms [CBB+05] and the usage of viewpoints conforms to the IEEE 1471 standard. Section 4.4.2, 4.4.3 and 4.4.4 describes how the viewpoints from Clements et al. and the views, which are chosen per viewpoint, address the concerns of the stakeholders.

4.4.2 Module viewpoint
The elements of the module viewpoint are the different modules that comprise an e-Learning solution where a module is an implementation of a piece of software that provides a coherent set of functionalities. Modules in this thesis are mapped to components that are part of an MLE.
A component in a module view has one of the following relationships [CBB+05]:
• *Is part of:* The is-part-of relationship defines part/whole relationship between two components.
• *Depends on:* The depends-on relationship defines dependencies between components.
• *Is a:* The is-a relationship defines a generalisation relationship between a more specific component and a more generic component.
Views that conform to the module viewpoint can use different architectural styles [CBB+05]. Two views that are important for this thesis are the Module Decomposition View and the Module Layer View. The Model Decomposition View shows how an e-Learning solution is decomposed into components or applications and how functionality is
allocated to those elements. The Module Layer View deals with the way a complex e-Learning architecture can be decomposed into abstract dimensions such as presentation, application logic, persistent data and infrastructure ([BHS07A], [AVZD05]).

Example: In an MLE the learning services could be decomposed into an LMS, LCMS, Assessment System and a Collaboration System component. These components are related to the learning services with an is-part-of relationship. Table 4.4-1 gives an overview of the stakeholders who are interested in module views and the concerns which are addressed by module views.

<table>
<thead>
<tr>
<th>Viewpoint name</th>
<th>Module viewpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stakeholders</td>
<td>Customer / Business owner; End user; Development team</td>
</tr>
<tr>
<td>Concerns</td>
<td>Module Decomposition View addresses the following concerns:</td>
</tr>
<tr>
<td></td>
<td>• What is the general idea behind the solution?</td>
</tr>
<tr>
<td></td>
<td>• What are the reporting capabilities of the solution?</td>
</tr>
<tr>
<td></td>
<td>• Are the functional requirements implemented in the solution?</td>
</tr>
<tr>
<td></td>
<td>• Does the solution provide end users relevant learning, when they need it in order to perform their jobs well?</td>
</tr>
<tr>
<td></td>
<td>• Does the solution have capabilities to link learning to skills and jobs or functions?</td>
</tr>
<tr>
<td></td>
<td>• Does the solution have reporting capabilities that can measure end users’ progress on their personal development plan and the effectiveness of learning programs?</td>
</tr>
<tr>
<td></td>
<td>• Is the solution easy to use; is it easy for end users to find their way through the different parts of the solution?</td>
</tr>
<tr>
<td></td>
<td>• In which components is the solution decomposed?</td>
</tr>
<tr>
<td></td>
<td>• How is the functionality allocated to the different components of the solution?</td>
</tr>
<tr>
<td></td>
<td>• How are non-functional requirements such as usability, security, flexibility, interoperability and maintainability supported?</td>
</tr>
<tr>
<td>Module Layer View addresses the following concerns:</td>
<td></td>
</tr>
<tr>
<td>• In which layers is the solution decomposed?</td>
<td></td>
</tr>
<tr>
<td>• How is the functionality allocated to the different components of the solution?</td>
<td></td>
</tr>
<tr>
<td>• How are the different components layered across the solution?</td>
<td></td>
</tr>
<tr>
<td>• How are non-functional requirements such as interoperability and scalability supported?</td>
<td></td>
</tr>
<tr>
<td>Rationale</td>
<td>The viewpoint gives a clear overview of the different components of an e-Learning solution and how these components interact with each other.</td>
</tr>
</tbody>
</table>

Table 4.4-1: Overview of the module view’s stakeholders and their concerns
4.4.3 Component and connector viewpoint

Component and connector views define models consisting of runtime entities. A runtime entity is an instance of a component and connector type, where component types are processing units and connector types are interaction mechanisms [CBB+05]. Examples of component types are processes, objects, clients, servers and data stores. Examples of interaction mechanisms are procedure calls between a client and a server or components that communicate with each other using a Publisher-Subscriber mechanism. Views that conform to the component and connector viewpoint can use a variety of architectural styles. Clements et al. describe six architectural styles: pipe-and-filter style, shared-data style, Publisher-Subscriber style, client-server style, peer-to-peer style and the communicating-process style [CBB+05]. Of these six styles the shared-data, Publisher-Subscriber and communicating-process style are important because they describe the runtime behaviour that can be expected in e-Learning solutions. Avgeriou and Zdun have created a pattern language that comprises eight component and connector views and patterns that are classified in each view [AVZD05]. The styles used by Clements et al. are similar to the patterns used by Avgeriou and Zdun. In this thesis the Data-centered View and the Communication Interaction View are used because these views describe the communication between the components in the business logic Layer and the communication between the components in the business logic Layer and the database Layer [AVZD05].

In the Data-centered View the solution is viewed as persistent (shared) data stores that are accessed and modified by components of the business logic Layer in an e-Learning solution. Each component has its own data store and the components are independent of each other. The elements that transfer data written or read from the data stores are connectors that are attached to the data stores and the components [AVZD05].

In the Communication Interaction View the solution is viewed as a number of independent components that interact with each other in the context of an e-Learning solution. The components are independent of each other because they only exchange data but do not control each other [AVZD05]. The components interact with each other through connectors using one of the following integration styles: File and Transfer, Remote Invocation Method, Publish and Subscribe or Messaging [BHS07a].

Example: In a MLE a Human Resource system (HR system) could be responsible for the user administration; the HR system could use the Messaging integration style to make all updates to user profiles available to the Learning Management system or other components in the business logic Layer.

Table 4.4-2 gives an overview of the stakeholders who are interested in components & connector views and the concerns which are addressed by these views.

<table>
<thead>
<tr>
<th>Viewpoint name</th>
<th>Component and connector viewpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stakeholders</td>
<td>Development team, Maintenance and Support team</td>
</tr>
</tbody>
</table>
Concerns | Communication Interaction View:
---|---
• What is the relationship between the components of the solution?
• What is the functionality allocated to the different components of the solution?
• How are the different components layered across the solution?
• Which integration styles are used in the solution?
• How do the independent components in the solution interact with each other?
• How are individual components decoupled from each other in the solution?
• How are the non-functional requirements such as performance, flexibility, security, interoperability and maintainability supported?

Data-centered View
• What data is stored by which components in the solution?
• How is the data created, accessed and updated by the different components in the solution?
• How are the non-functional requirements such as availability, performance, security, and maintainability supported?

Rationale
The viewpoint gives a clear overview of the interfaces between the different components of an e-Learning solution, how these interfaces interact with each other and what interaction mechanisms are used by the interfaces.

Table 4.4-2: Overview of the component and connector view’s stakeholders and their concerns

4.4.4 Allocation viewpoint
The relation used in the allocation viewpoint is the *allocate-to* relationship which allocates software elements to environmental elements. Clements et al. define three different views: the deployment view, the implementation view and the work assignment view. For the reference architecture for e-Learning solutions only the deployment view is important because the view maps software units to processing nodes, communication channels and data storage.

Example: In an MLE the Assessment System could be allocated to a server with its database allocated on a different server. The communication link between the Assessment system and the database uses ODBC or JDBC.

Table 4.4-3 gives an overview of the stakeholders who are interested in deployment views and the concerns which are addressed by deployment views.

<table>
<thead>
<tr>
<th>Viewpoint name</th>
<th>Allocation viewpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stakeholders</td>
<td>Development team, Maintenance and Support team</td>
</tr>
</tbody>
</table>
| Concerns       | • What is the system topology with respect to computational nodes on the network?  
|                | • What network protocols are used by the different nodes |
on the network?
- Which component is installed on which node on the network?

| Rationale       | The viewpoint is selected because the deployment style gives a clear overview of the topology of nodes on the network, which protocols are used and which component resides on a node. |

Table 4.4-3: Overview of the allocation view’s stakeholders and their concerns

4.5 Architectural views

4.5.1 Introduction

This subsection gives an overview of the views used to design the architecture and the elements of these views and how these elements are mapped to one another. Each view will be documented using a form as described in section 3.4.3. and will also document the architectural background using architectural patterns.

Module Layer View: The elements of this view are the layers in which the infrastructure subsystem of an MLE can be decomposed. The design of an MLE uses a triple-layer classic web application architecture. The benefits of this architecture are that it is universally acknowledged and the independence of the three layers ensures platform neutrality and exchangeability of the components within each layer ([BMR+96], [DYLO04], [Fow03]).

Module Decomposition View: The elements of this view are the systems of the business logic layer of an MLE. The systems of the presentation layer and the database layer are not taken into account because the focus is on integration between the systems that comprise the business logic layer. The decomposition of an MLE into systems is rather coarse-grained because these systems are the building blocks BT LS uses to integrate with one another.

Component Interaction View: The Component Interaction View is subdivided into four view packets:
- Component Interaction view packet 1: File Transfer. The elements of this view packet are the File Transfer processes that take place between systems of the Module Decomposition View.
- Component Interaction view packet 2: Messaging. The elements of this view packet are Messaging processes between several systems of the Modules Decomposition View.
- Component Interaction view packet 3: Remote Procedure Invocation. The elements of this view packet are the API calls that take place between systems of the Module Decomposition View.
- Component Interaction view packet 4: Publish-Subscribe. The elements of this view packet are the different Events that take place between the systems of the Module Decomposition View.

Data-centered View: The elements of this view are the database interfaces of the systems that comprise the Application Server Architecture. This view only describes the database interfaces of the core applications of an MLE and not the database interfaces that are part of
the common infrastructure. The reason for that BT LS is only responsible for the design of the Application Server Architecture.

**Deployment View:** The elements of this view are the systems that are part of the three layers and how there are mapped onto network nodes and the protocols they use to communicate with one another.

### 4.5.2 Module Layer View

In the Module Layer View the infrastructure subsystem of an MLE is seen as an enterprise wide environment that can be decomposed into interacting parts.

#### Primary presentation

![Layers for a MLE](image)

<table>
<thead>
<tr>
<th>Name</th>
<th>Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrastructure subsystem of an MLE</td>
<td>Layer 1: Presentation</td>
</tr>
<tr>
<td></td>
<td>Layer 2: Business logic</td>
</tr>
<tr>
<td></td>
<td>Layer 3: Database</td>
</tr>
</tbody>
</table>

Table 4.5-1: Primary presentation module layer view

The diagram and the table show the decomposition of the infrastructure of an MLE into three layers: presentation layer; business logic layer and the database layer. The arrows between the layers, in figure 4.5-1, point to a bi-directional communication between the layers.

#### Element catalogue:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Element Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presentation Layer</td>
<td>Presentation Layer consists of one component the portal. The portal gives a single port of entry for the users and handles the interaction between the users and the different components in the business logic layer. Access to the components is based on a user-profile.</td>
</tr>
<tr>
<td>Business logic Layer</td>
<td>The business logic of an e-Learning solution is responsible for all the work an e-Learning solution is supposed to do. The business logic layer is decomposed into two service layers: the common services and the learning services. Common Services provide functionality such as mail, authentication of users that is used by other systems. The Learning Services provide the core functionality for an e-Learning solution such as LMS functionality or LCMS functionality.</td>
</tr>
<tr>
<td>Database Layer</td>
<td>This layer is responsible for the communication with the databases of the different components.</td>
</tr>
</tbody>
</table>
The relationship in the layer view is *allowed to use* or *uses*. The software components in a layer are allowed to use or uses software components from the same layer or any other layer immediately below or adjacent to the layer.

**Context diagram**

The context for the module layer view is the Domain Model for e-learning solutions described in section 4.1.2. The model for an MLE shown in figure 3.2-3 serves as the context diagram for this module layer view.

**Architectural background**

When transforming a Domain Model for an e-Learning solution into a technical architecture it should be possible to develop and evolve different system parts independently. The challenge is to find a balance between a design that partitions an e-Learning solution into software components that can be developed independently but does not lose sight of the architectural vision of the solution. Introducing layers into an architecture improves independent development and evolvement of system’s functionality. Layering is one of the most commonly used architectural structures that reflect a division of an e-Learning solution into software units ([CBB+05], [BHS07a], [Fow03]). The layers of an e-Learning solution are based on the infrastructure subsystem of an MLE where an MLE is an enterprise wide environment embedded in the infrastructure of an organisation. The infrastructure subsystem of a MLE is based on the UKeU framework for e-Learning solutions (see section 3.1.2). The layers in the UKeU framework describe a logical structure of an architecture for an e-Learning solution. The services in the UKeU framework are grouped into four different layers:

- **Layer 1: Portal**
- **Layer 2: Common Services**
- **Layer 3: Learning Services**
- **Layer 4: Databases**

Layer2- and layer-3 in the UKeU framework are divided into segments denoting a decomposition of the layer into software components. Each of the components in layer-2 or layer-3 has its own interface for communication with other components. Layers in the UKeU framework do not separate higher-level from lower level responsibilities. Layer-3 in the UKeU framework describes the services that produce and consume e-Learning resources whereas the services in layer-2 are needed by everyone regardless of their role. The database layer in the UKeU is used by all software components in layer-2 and layer-3. The logical layering of the UKeU framework is transformed into Layers ([CBB+05], [BHS07a], [Fow03]) of an architecture for a MLE. The Layers of the architecture comprises three principal Layers: the portal layer, the business logic layer and the database layer. These Layers are the similar to the principal Layers for enterprise applications as discussed by Fowler [Fow03] and the Layers for information systems as discussed by Dyson and Longshaw [DYLO04].

The presentation layer handles the interaction between a user and the systems that comprise the business logic layer. The presentation layer consists of an HTML based browser user interface and displays information to the user and interpreted commands from the user into actions upon the business logic layer components [Fow03]. The presentation layer is not specified in much more details because portal is in most cases one of the enterprise
applications with which the components of an MLE have to communicate and is therefore not part of an e-Learning solution.

The business logic layer is divided into layers and has components within a common service layer and within a learning services layer. The components of a learning service layer have to be able to receive requests from the presentation layer and need to be able to respond to those requests. The communication between the components in the learning services and the presentation layer is done through services that are defined for each of the components that comprise the learning services. The services can either use Remote Procedure Invocation or the Publisher-Subscriber approach for their communication [Fow03]. The components of the common services part of the business logic layer serve as supporting components of the learning services and do not communicate directly with the presentation layer.

The database layer’s main responsibility is storing persistent data. Another important part is the communication with components of the business logic layer. The latter is a bi-directional communication where a component sends SQL requests to data sources and data sources send responses back to components. The interface between components of the database layer and the business logic layer is described in more details in the Data-centered View.

4.5.3 Module Decomposition View

The layer view described in section 4.5.2 shows the different layers of the infrastructure subsystem of an MLE. The module decomposition view decomposes the business logic into smaller, strictly separated modular parts which each have clearly defined and scoped responsibility.

Primary presentation

Figure 4.5-2 displays the graphical representation of the components of the business logic layer. The components are grouped into two services, the Common Services and the Learning Technology Services. The Common Services consist of more generic components of an e-Learning solution whereas the Learning Technology Services consist of the specific learning technology components.

One application is missing in the diagram and that is the mail application because all applications use the mail application to forward their notifications.

Figure 4.5-2 also shows the relationships between the different systems and also the direction of the relationship. The arrow direction points to the dependency of the systems i.e. The LMS has a “uses” interface with the LCMS which means that for a proper working it needs data from the LCMS. This does not mean that the LMS cannot work without the LCMS but in case of access to content and progress information of learners it needs data from the LCMS.
Element catalogue

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Element Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Management System</td>
<td>A user management system manages users, groups and roles from all components that comprise the solution. The following key functions are relevant for this reference architecture:</td>
</tr>
<tr>
<td></td>
<td>• User registration and account management</td>
</tr>
<tr>
<td></td>
<td>• Authentication management</td>
</tr>
<tr>
<td></td>
<td>• Authorisation management</td>
</tr>
<tr>
<td>Mail System</td>
<td>A mail system is responsible for sending, retrieving and forwarding e-mails or notifications from the different components that comprise the solution.</td>
</tr>
<tr>
<td>HR-system</td>
<td>A human resource system (HR-system) merges Human Resource Management as a discipline, its particular basic HR activities and its HR processes with the information technology field. The following key functions are relevant for this reference architecture:</td>
</tr>
<tr>
<td></td>
<td>• End user profile management including skills, competencies and job</td>
</tr>
<tr>
<td></td>
<td>• Creating and maintaining end user personal development plans and personal development records</td>
</tr>
<tr>
<td>Collaboration System</td>
<td>Collaboration System ([CENT], [WEBX])</td>
</tr>
<tr>
<td></td>
<td>A collaboration system supports the use of collaborative tools</td>
</tr>
<tr>
<td>Element Name</td>
<td>Element Responsibilities</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>and provides functionality to create and manage collaborative sessions including asynchronous and synchronous communication. A collaborative system has the following key functions:</td>
</tr>
<tr>
<td></td>
<td>• Managing and set-up of a collaborative session</td>
</tr>
<tr>
<td></td>
<td>• Supporting synchronous communication (video conferencing, virtual classrooms, application sharing, shared white board)</td>
</tr>
<tr>
<td></td>
<td>• Supporting asynchronous communication (wiki, blog, discussion groups, e-mail group)</td>
</tr>
<tr>
<td>Learning Management System</td>
<td>Learning Management System (LMS) ([Mood], [SABA], [SUMT])</td>
</tr>
<tr>
<td></td>
<td>An LMS is an application that automates the administration of learning events and has the following key:</td>
</tr>
<tr>
<td></td>
<td>• End-user user account details</td>
</tr>
<tr>
<td></td>
<td>• Import and publishing of AICC and SCORM content</td>
</tr>
<tr>
<td></td>
<td>• Course management</td>
</tr>
<tr>
<td></td>
<td>• Catalogue management</td>
</tr>
<tr>
<td></td>
<td>• Enrolment management</td>
</tr>
<tr>
<td></td>
<td>• Resource management</td>
</tr>
<tr>
<td></td>
<td>• Competency / skills management</td>
</tr>
<tr>
<td></td>
<td>• Tracks user progress on courses</td>
</tr>
<tr>
<td></td>
<td>• Providing reports</td>
</tr>
<tr>
<td>Learning Content Management System</td>
<td>Learning Content Management System (LCMS) ([EEDO], [GIUN], [OUTS])</td>
</tr>
<tr>
<td></td>
<td>An LCMS is an application that enables authors to register, store assemble, manage and publish learning content for delivery via the web and has the following key:</td>
</tr>
<tr>
<td></td>
<td>• Content management</td>
</tr>
<tr>
<td></td>
<td>• Content authoring</td>
</tr>
<tr>
<td></td>
<td>• Export content into AICC or SCORM packages</td>
</tr>
<tr>
<td></td>
<td>• Import SCORM packages</td>
</tr>
<tr>
<td>Assessment system</td>
<td>Assessment System [QMP]</td>
</tr>
<tr>
<td></td>
<td>An assessment system is an application that enables authors to create surveys, formative assessments and summative assessments for delivery via the web and has the following key functions:</td>
</tr>
<tr>
<td></td>
<td>• Storing, retrieving, deleting and updating assessments</td>
</tr>
<tr>
<td></td>
<td>• Export or import an QTI assessment</td>
</tr>
<tr>
<td></td>
<td>• Export an assessment as AICC or SCORM package</td>
</tr>
</tbody>
</table>

Table 4.5-3: Element catalogue module decomposition view

The decomposition of the business logic results in seven components and is part of relationship between the components. The decomposition also shows the relationships between the applications.
Context diagram
The context for the Module Decomposition View is the layered structure shown in figure 4.5-1 and the model for an MLE shown in figure 3.2-3. Figure 4.5-1 shows the business logic layer and the two services components and figure 3.2-3 shows the components that comprise each of the service components.

Architectural background
The Domain Model is decomposed into Domain Objects where each Domain Object encapsulates all the functions of one application from the business logic layer. The Domain Objects capture business or infrastructure functionality. One of the major challenges when decomposing Layers into Domain Objects is avoiding tight connections between Domain Objects. Tight connections between Domain Objects increase the coupling of Domain Objects because changes in the implementation of any of the Domain Objects can have an effect on all other Domain Objects. Loose coupling between Domain Objects is achieved by letting a client of a Domain Object access its functionality only through an explicit interface [BHS07a].

Two Domain Objects in the business Layer are the LMS and the LCMS. Each object has its own functionality and responsibilities as described by the functional requirements described in section 4.3.2. SumTotal or Saba are examples of an LMS whereas Outstart or ForceTen are examples of an LCMS.

The decomposition of the Domain Model into Domain Objects where each Domain Object encapsulates all the functions of an application is a coarse-grained decomposition. The reason this coarse grained decomposition is that BT LS develops solutions for their customers based on COTS applications or open source applications. A solution provided by BT LS integrates the COTS applications or open source applications into one solution for the customer.

The Application Server Architecture and Peripheral Specialist Elements patterns are used to separate the Domain Objects into objects that provide the core functions of an e-Learning solution and objects that are not part of the mainstream functions. The Domain Objects that form part of the Learning Services provide the core functions of an e-Learning solution and are therefore all part of the Application Server Architecture. The Domain Objects that form part of the Common Services are not part of the mainstream functions of an e-Learning solution and are therefore part of the Peripheral Specialist Elements.

Figure 4.5-3 gives an example of the decomposition of the business logic layer into an LMS, an LCMS, a HR-system and a Mail system as the Domain Objects. The LMS and LCMS are part of the Application Server Architecture and the HR-system and the Mail system are part of the Peripheral Specialist Elements. Figure 4.5-3 also shows that user account and user preference information from the LMS to the LCMS and content status data is send from the LCMS to the LMS. The data exchange between the LMS and LCMS is done through a SCORM interface that is implemented by the LMS. The integration style used for this integration between an LMS and an LCMS is described in more details in Communication Interaction View.
An Application Server Architecture provides the foundation of the Learning Services where quality attributes such as availability and performance can be addressed more easily [DYLO04]. There is always a trade off between the functionality a customer would like to group into one system and which functionality a customer would like to have in different systems. Having multiple systems where some systems could continue to work where others fail or have been switched off is a real benefit [DYLO04]. Another benefit of having multiple systems is that businesses are not bound to one supplier. On the other side the performance of a solution where the core functionality is grouped together into one system is slightly better compared with a system that has functionality mapped to different sub-systems [DYLO04].

In this thesis the Application Server Architecture pattern [DYLO04] is used to map each functional requirement as described in section 4.3.2 to a key function of one the applications that comprise the Application Server Architecture.

The functional and non-functional characteristics of functions that are not part of the ‘mainstream’ functionality of an e-Learning solution often differ from the characteristics that belong to the components that form part of the core solution [DYLO04]. Having implemented an e-Learning solution using an Application Server Architecture makes the solution easier to manage, but having all relevant functionality available from components of an e-Learning solution can make the solution highly complex and difficult to create. Furthermore, the ‘extraordinary’ functionality is often created, managed, maintained and used by other parts of a business. Different components of a MLE, such as an LMS or an LCMS have e-mail capabilities to send notifications to users of the components. User profile information is normally managed by HR-systems such as PeopleSoft and is used by other systems such as an LMS. Mail systems, HR-systems are not part of the ‘mainstream’
functionality of an e-Learning solution and should therefore be added to the periphery of an e-Learning solution architecture.

Products such as TotalLMS, Saba LMS, Evolution and Learn Exact ([SABAa], [SUMTb], [BRHA05a], [BRHA05b]), all use e-mail to send notifications to users of the systems. None of these products have added complete e-mail functionality to their application but they all offer the option to integrate the application with a mail system (SMTP server). In most cases a mail system is part of a corporate business infrastructure and therefore not part of the architecture of an e-Learning solution. These products also provide communication mechanisms to exchange data with other systems such as HR-systems. By adding Domain Objects that do not belong to the core part of an e-Learning solution to Peripheral Specialist Elements improves the availability of the of the architecture of an e-Learning solution because these specialist functions are not part of ‘mainstream’ functionality of the architecture for e-Learning solutions [DYLO04]. The overall performance is also improved because these ‘extraordinary’ tasks are carried out by other systems on the infrastructure of a corporate business, which run on their own dedicated servers [DYLO04]. Peripheral Specialist Elements improve the usability of the architecture of an e-Learning solution because it can deliver non-core functionality in a flexible manner [DYLO04]. The overall security of the architecture for an e-Learning solution could become more challenging because Peripheral Specialist Elements introduce more integration or communication points to the architecture. However, the security these elements could be individually tailored by separating these elements’ security from the core part of the architecture [DYLO04]. By splitting up specialist behaviour and (re-)using existing systems to provide this specialist functionality saves development, hardware and maintenance costs [DYLO04].

4.5.4 Component Interaction View

The Component Interaction View focuses on the communication between the components in the business logic Layer of an MLE. A component can have multiple connectors and each connector uses one of the following integration styles: File Transfer, Messaging, Remote Procedure Invocation or Publisher-Subscriber ([HOWO04], [BCK07a]). To break down the complexity of the view into more understandable chunks it is decomposed into several view packets. Each view packet focuses on one of the integration styles that integrate the components to one another.

Component Integration view packet 1: File Transfer

Primary presentation

![Figure 4.5-4: Primary presentation File Transfer integration style](image)
Figure 4.5-4 gives an overview of the process that is involved to export a file from an LMS to a HR-system. The figure shows the LMS, the export interface, the file that is transported, the import interface and the HR-system. Similar processes could be used to exchange data between two systems by exporting and importing files.

**Element catalogue**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Type</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMS_Transcript_Export process</td>
<td>process</td>
<td>This element is responsible for extracting the transcript data from a LMS database and to create an export file. The export file is transported by the network infrastructure and send a HR-system for reporting purposes.</td>
</tr>
</tbody>
</table>

Table 4.5-4: Element Catalogue component interaction view-packet 1: File Transfer

The relation type used in this view packet is: *attachment*. The primary presentation of this view packet shows how the two components and related connectors are attached to one another.

**Context diagram**

![Context diagram](image)

Figure 4.5-5: High Level overview of data flow between the different layers of an MLE

Figure 4.5-5 gives an overview of the basic flow of data exchanged between the different layers of an MLE. The focus in the Component Interaction View is on the integration styles used by the applications that form part of the business logic layer.

**Architecture background**

File Transfer is used as integration style when a pre-defined file format is agreed for sharing data between two applications and when it is acceptable that two applications are not in sync with one another [HOWO04]. File Transfer is a simple way of exchanging data between applications because no extra tools or integration packages are needed, but this also implies that a developer has to do a lot of the work himself [HOWO04]. The developers of the interface have to agree on file-name conventions and directories where the files will appear
and should add functionality to the interface that takes care of the old files. A time schedule should be agreed when an application produces the file and when it consumes the file. An LMS tracks data about all learning, such as enrolments and completions of learning activities that take place in a company whereas in a HR-system a company reports on all learning that took place. For reporting purposes it is not required that both systems are in sync. The File Transfer between the two systems is designed as shown in more detail in figure 4.5-6. An SQL process (LMS_Transcript_Export process) is initiated on the database of the LMS to extract the learning history out of the database and to create an export file with the data. A file transfer process (FTP, Secure FTP) is scheduled to search for the file and to send the file to the HR-system. On the HR-system a process (Import Transcript Data process) runs to import and process the data in its database. After the completion of this process the data is available for reporting purposes by the HR-system. The file format used to create the file could either be CSV (Comma Separated) or XML and is created based on an agreed time schedule (once a day or week). Organisations normally schedule these tasks in out of office hours when the systems are less used.

The LMS_Transcript_Export process is a process that is part of the Application Server Architecture of an MLE whereas the Import Transcript data process is a process that is part of the Peripheral Specialist Elements of an MLE. Products such as TotalLMS or Saba LMS have processes in place which can be used to import or export data ([SABAc], [SUMTc]). The process is recommended, by the suppliers, for the import or export of large files. The transfer process is something a developer has to set up using the existing hardware infrastructure.

**Component Integration view packet 2: Messaging**

**Primary presentation**

Figure 4.5-7: Primary presentation Messaging integration style
Figure 4.5-7 gives a high level overview of the process where components are integrated to one another by a Messaging System. The figure shows the systems that are connected to a Publish-Subscribe channel. The arrows details whether a system pulls data from the channel or push data to the channel. The connectors attached to a component are using messages to exchange data.

**Element catalogue**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human_Resource_Sytem_Updates</td>
<td>Process</td>
<td>This element is responsible for adding messages to a message queue. These messages are read by the User Management System.</td>
</tr>
<tr>
<td>User_Management_Updates</td>
<td>Process</td>
<td>This element is responsible for reading messages from a message queue and updates the user profiles in the User Management System database. An administrator from the User Management System is responsible for adding roles and access permissions to user profiles.</td>
</tr>
<tr>
<td>LMS_User_Updates</td>
<td>Process</td>
<td>This element is responsible for reading messages from a message queue and updates the user profiles in the LMS database. User roles and access permissions are part of the messages retrieved from the User Management System. Every user will need to have access to the LMS where the majority of the user will have a role as Student.</td>
</tr>
<tr>
<td>LCMS_User_Updates</td>
<td>Process</td>
<td>This element is responsible for reading messages from a message queue and updates the user profiles in the LCMS database. User roles and access permissions are part of the messages retrieved from the User Management System. Only End-Users responsible for the content creation and maintenance will have access to the LCMS.</td>
</tr>
<tr>
<td>AssessmentSystem_User_Updates</td>
<td>Process</td>
<td>This element is responsible for reading messages from a message queue and updates the user profiles in the Assessment System database. User roles and access permissions are part of the messages retrieved from the User Management System.</td>
</tr>
</tbody>
</table>
The relation type used in this view packet is: attachment. The primary presentation of this view packet shows how the components and related connectors are attached to each other.

**Context diagram**  
The context diagram for this view packet is shown in figure 4.5-5.

**Architecture background**  
Asynchronous Messaging is fundamentally a pragmatic reaction to the problems distributed systems face. Sending messages does not require the systems to be up and ready to be used at the same time [HOWO04]. Asynchronous communication between distributed systems force developers to recognise that the overall performance is slower compared to a monolithic system which will encourage developers to design the components with high cohesion and low adhesion. Messaging systems provide the same decoupling as File Transfer but with the advantage that it allows integrators to choose between broadcasting messages to one or more receivers [HOWO04].

Components of an e-Learning solution use a Publisher-Subscriber Message Channel that connects the sending components with one or more receiving components. A Publish – Subscribe Message Channel has one input channel that splits into multiple output channels, one for each receiving component. A Publisher-Subscriber Message Channel delivers a copy of the Message to each of its output channels where each output channel has a Message Endpoint that allows the receiver to read the message only once ([HOWO04], [BHS07a]). To avoid eavesdropping on the message channel, subscribing to a channel, should be restricted by security policies.

The Messages between the connecting components have a predefined and agreed format similar to the files in the File Transfer integration style. This could mean that developers of the subscriber application have to add functionality to their interface to translate the data format used in the message into a data format that is readable by the subscriber application. Figure 4.5-8 gives an overview of the communication between the User Management System and the LMS, LCMS and the Assessment System. The User Management System connects to the Message Channel, such as IBM’s MQ-Series, through a Message Endpoint. The Message Endpoint is a client, such as the IBM MQ client, installed on a network node on a corporate network. The User_Management_Updates process extracts the changes made to user profiles from the User Management database and produces an XML message which is added to Message Channel. The receivers’ processes, such as the LMS_User_Updates or the LCMS_User_Updates, read and process the XML messages in their database.
None of the LMS applications, such as TotalLMS or Saba LMS, or the LCMS applications, such as Evolution or ForceTen, have an out of the box solution for the communication with a Messaging System ([EEDOa], [OUTSa], [SABAb], [SUMTb]). Most of these products do have a data import web service which can be used to connect a LMS or a LCMS for the import of data from a Messaging System.

To integrate with a Message System developers have to write code that reads a Message received through a Message Endpoint and calls the LMS web service to import data. The advantage of implementing messaging solution, instead of using a web service solution, is that the communication between sender and receivers is asynchronously and a sender does not know anything of the receivers of the messages. Another advantage is that the format of the messages is not defined upfront but can be transformed at the receiver side. The latter means that systems are much more decoupled from one another than with other integration styles [HOWO04].

Using a messaging system to integrate systems with each other has a positive impact on the flexibility of an e-Learning solution but because of the higher degree of decoupling of the systems. Using a messaging system has a negative impact on the maintenance and costs of an e-Learning solution because of the existing Learning Technology systems do not have out-of-the box connectors to connect to a Message Channel.

**Component Integration view packet 3: Remote Procedure Invocation**

**Primary Presentation**

Figure 4-5.9 shows the Remote Procedure Invocation integration style between an LMS and LCMS. A course structure of an online course is stored in the LMS schema and the actual physical files of the course are stored in the LCMS schema. The LMS uses its SCORM
interface to launch online content from a LCMS (1) and the LCMS sends status updates back to the LMS (2) using the SCORM interface of the LMS.

### Component catalogue

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMS_getValue</td>
<td>API Call</td>
<td>LMS_getValue is part of the SCORM / AICC API which is implemented by every LMS vendor that is compliant with the SCORM / AICC specification. When a user launches a SCO, from a LCMS or Assessment system, it sends a getValue API call to the LMS for retrieving data, such as learner first / last name or suspended data.</td>
</tr>
<tr>
<td>LMS_setValue</td>
<td>API Call</td>
<td>LMS_setValue is part of the SCORM / AICC API which is implemented by every LMS vendor that is compliant with the SCORM / AICC specification. When a user exits a SCO, in a LCMS or Assessment system, it sends a setValue API call to the LMS for storing tracking data, such as score information or progress towards completion of the SCO.</td>
</tr>
</tbody>
</table>

Table 4.5-6: Element catalogue component interaction view-packet 3: Remote Procedure Invocation

The relation type used in this view packet is: *attachment*. The primary presentation of this view-packet shows how the components (LMS, LCMS) and related connectors (LMS_getValue, LMS_setValue) are attached to one another. The same diagram could be used to attach LMS_getValue and LMS_setValue to the Assessment system.

### Context diagram

The context for this view packet is shown in figure 4.5-5.

### Architectural background

Remote Procedure Invocation applies the principle of encapsulation to integrate systems. If an LCMS or Assessment system needs data that is owned by an LMS it sends API calls to ask the LMS directly. Similarly if an LCMS or Assessment system needs to modify data that is owned by an LMS it does so by making API calls to the LMS. This allows each system to maintain data that they own. The systems are compliant with the AICC / SCORM specification so an LCMS or an Assessment system knows which API calls and data elements are supported and can be used to retrieve or store data. Encapsulation and usage of standard specifications such as SCORM or AICC specification help reduce the coupling between systems.

Remote Procedure Invocation, compared with local procedure calls, can lead to slow and unreliable systems because it depends on the underlying hardware infrastructure that is used.
for the communication between the systems [HOWO04]. Therefore, the usage of Remote Procedure Invocation should be minimised.

Both the Evolution LCMS and the ForceTen LCMS provide an option for dynamic delivery of content by using the AICC standard. In the dynamic delivery the ForceTen or Evolution courses are exported as links back into the ForceTen server; the exported course contains no content. This means that when the course is launched through the TotalLMS or Saba LMS, the content is actually coming directly from the Evolution or ForceTen server. Since the course content is created dynamically, any changes to courses in Evolution or ForceTen are reflected to the user as soon as they happen. Thus, if a graphic is changed, all the courses using the graphic see the change immediately with no intervention from the system administrator, and if a module changes, the change is reflected in all courses using the module. Figure 4.5-10 gives an overview of the dynamic course delivery process between an end-user’s browser, the TotalLMS or Saba LMS and the Evolution or ForceTen LCMS. The end-user sends a request for content to the LMS (1), the LMS forwards the request to the LCMS (2), the LCMS runs the content in the end-user’s browser (3), the end-user’s browser sends progress information back to the LCMS (4), the LCMS sends status information to the LMS (5) and the LMS shows the status information in the end-user’s browser (6).

![Diagram of Dynamic Course Delivery](image)

Question Mark Perception (QMP) also provides an option to launch an assessment from a LMS using the AICC / SCORM specification [QMPb]. These steps are similar to the ones that are taken when a SCO is launched from an LCMS as shown in figure 4.5-10.

Web based training and online learning material run in end user’s web browsers, such as Internet Explorer or Netscape. As a security measure web browsers do not allow content
from one place to communicate with content coming from a different place. This prevents potentially malicious practices such as web pages from one company to assessing data entered from another company. This also prevents content launched from an LMS to communicate with learning objects stored in a LCMS repository. This so called cross-domain security problem is caused by content created according to the AICC or SCORM specifications. Suppliers such as SumTotal, Saba or Outstart all have solutions in place to overcome this cross-domain security problem ([OUTSa], [SABAb], [SUMTb]). These suppliers add a so called web plug-in, on the servers with the actual physical content files, that establish direct communication between the content server and the LMS.

**Component Integration view packet 4: Publisher-Subscriber**

In view packet 2 the Publisher-Subscriber pattern was used to integrate learning technology components through a Messaging System. In this view packet the Publisher-Subscriber pattern will be used to integrate learning technology components through web services.

**Primary presentation**

![Diagram of primary presentation Publisher-Subscriber integration style](image)

Figure 4.5-11 shows three applications, a portal, an LMS and a collaboration system. These applications are connected to the web that is responsible for sending the SOAP messages over HTTP. The arrows point to the direction of the communication. The portal is only receiving SOAP messages and the LMS and the collaboration system are sending and receiving SOAP messages.

**Element catalogue**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMS_GetActivity</td>
<td>Event</td>
<td>The Get Activity service handles requests from a Portal to get learning offerings information, based on end user preference settings, from a LMS.</td>
</tr>
<tr>
<td>LMS_CatalogueSearch</td>
<td>Event</td>
<td>The Catalogue Search service handles search requests from an end user and sends messages to the Portal that meet the passed specifications.</td>
</tr>
<tr>
<td>LMS_ResultReporting</td>
<td>Event</td>
<td>The Result Reporting service</td>
</tr>
<tr>
<td>Element Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Collaboration_Offering</td>
<td>Event</td>
<td>The Offering service handles LMS requests to create, modify or delete virtual learning offerings in the Collaboration System.</td>
</tr>
<tr>
<td>Collaboration_UserAccount</td>
<td>Event</td>
<td>The UserAccount service handles LMS requests to create modify or delete user account information in the Collaboration System.</td>
</tr>
<tr>
<td>Collaboration_Registration</td>
<td>Event</td>
<td>The Registration service handles LMS requests to register or unregister end users and instructors for a virtual learning offering in a Collaboration System. This service also updates the existing registration information.</td>
</tr>
<tr>
<td>Authentication</td>
<td>Event</td>
<td>The Authentication Service handles the authentication requests sent to the LMS. It is passed log-on credentials for a user and returns an authentication certificate or token.</td>
</tr>
</tbody>
</table>

Table 4.5-7: Element catalogue component interaction view-packet 4: Publisher-Subscriber

The relation type used in this view packet is: *attachment*. The primary presentation of this view-packet shows how the components and related connectors are attached to one another.

**Context background**
The context for this view packet is shown in figure 4.5-5.

**Architectural background**
The Publisher-Subscriber integration style allows subscribers to register for specific events and producers to publish certain events that reach a specific number of consumers. Publisher-Subscriber, like Messaging, supports asynchronous communication, in which publishers send data to subscribers without blocking to wait for a response. Asynchronous communication decouples publishers and subscribers so that they can be active and available at different points in time, which is a great advantage in a distributed e-Learning environment. The Publisher-Subscriber integration style can be used by a Messaging System as described in the view packet 2. In this view packet the focus is on using the Publisher-Subscriber integration style in a web service environment where publishers and subscribers use an XML based protocol, Simple Object Access Protocol (SOAP), for sending or retrieving messages. The communication between publishers and subscribers is done over the network infrastructure using HTTP or HTTPS. The great advantage of using Internet protocols, such as HTTP and HTTPS, is that they are accepted protocols by the Internet community. The Web Services Definition Language (WSDL), provided by suppliers, is used by developers to implement web services in an e-Learning solution. The parameters used by all the web services are specified in the WSDL specification in the web
service SDK of the suppliers. Web services are useful for ad-hoc data retrieval, portal integrations, and small real-time data synchronisations.

In this reference architecture, web services are used to provide a personalised portal to the end users of an e-Learning solution. Web services from an LMS will update information to the catalogue or news page, in the portal, based on the preferences set by the user. Products, such as Saba or SumTotal, provide web services that enable developers to integrate the LMS with a portal ([SABAe], [SUMTc]).

One of the available web services is an LMS Get Activity web service that a Portal uses to retrieve update messages from the LMS regarding the learning offerings in the catalogue. This web service returns an XML message with all learning offerings data that meet the passed end user preferences. The Portal will show this information to the end user in a readable format. Another web service that is provided by Saba or SumTotal is the Catalogue Search Service. This web service provides the Portal with functionality that an end user can use to search for learning offerings in the catalogue of the LMS. The search service returns an XML message with all the offerings that meet the passed specifications. The Portal will show the end user the retrieved result in a readable format.

Web services are also used to integrate an LMS with a Collaboration System, such as the integration of SumTotal with WebEx or Saba with Centra. The LMS is responsible for managing learning offerings, learning events and registering of end users to events. The Collaboration System enables the end users and instructors to link together and participate in a live session. A Collaboration System is also responsible for the lower level details as streaming audio and video to the end users, letting end users send questions to the instructor and so on. The three services, described in table 4.5-4, are the supported web services that are important for this reference architecture. These web services allow an LMS administrator to create a new learning offering, add new users and register users to a learning offering.

The LMS sends SOAP messages to the collaboration system using the Offering, User Account or the Registration web services provided by the Collaboration System. The collaboration system sends a SOAP message to the LMS using the Result web service of the LMS.

Figure 4-5.12 shows the architecture detail of the integration of an LMS with a Collaboration System. A similar approach could be used for integrating a CMS, such as Moodle, with a LMS such as Saba or SumTotal ([SABAd], [SUMTe]). The three web services provide an LMS administrator with the functionality to create user accounts, to register end users to learning offerings and to manage learning offerings on Moodle. Moodle in such a solution is seen as a collaboration environment associated to a course. The LMS-Result_Reporting is the web service which is used by Moodle to mark a course as completed.
Single Sign On (SSO) is another area where web services are used to provide end users access to learning technology components such as an LMS, LCMS or an Assessment System. SSO is one of the techniques used for the Identification and Authentication (I&A) of users in an enterprise wide environment [MED+06]. SSO is one common approach to minimize time and effort in the context of an enterprise network by means of a single authentication that is performed when users initially access the network [MED+06].

A Front Door is used to provide an SSO mechanism to all learning technology components that comprise the e-Learning solution. An LDAP system is a popular solution for implementing Front Door in an enterprise wide environment [MED+06]. Using a Front Door means that an LDAP system will handle access control to the LMS, but authorisation ([MED+06], [Wei05]) will be performed by the LMS. LMS systems such as SumTotal or Saba provide a Role-Based Access Control ([MED+06], [Wei05]) model to assign rights on functions or tasks of end users in an LMS. Suppliers such as SumTotal, Saba, QMP or Outstart provide integrations with LDAP systems ([SABAb], [SUMTb]). In this thesis a LDAP system is part of the Peripheral Specialist Elements of a reference architecture for e-Learning solution.

Figure 4-5.13 shows the SSO implementation of an LMS. An end user wants to connect to the LMS through the portal. The LDAP agent will initially provide the user with a session when valid logon credentials have been supplied. In the end-user experience this session can be used by all web applications integrated by the LDAP system, including the LMS. In the backend all applications will use their own session management. When a user wants to access the LMS the request is intercepted by LDAP agent. The LDAP system checks if the user already has an active session. If not, the user is required to log into the LDAP system. The request is forwarded to the LMS, which needs to capture the credentials stored in the request and automatically provide the user with a LMS session. This ensures that the user only needs to supply logon credentials to LDAP system.
All learning technology components are protected by the Front Door and require SSO before an end user is able to access and use the component. The SSO implementation with an LCMS or Assessment System is similar to the SSO implementation with an LMS.

4.5.5 Data-centered view

In the Data-centered view an e-Learning solution is viewed as (shared) data stores that are accessed and modified by Learning Technology Components.

**Primary presentation**

Each of the Learning Technology Components has its own database schema and interface that connects the application logic of the component to its database.

**Element catalogue**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMS database interface</td>
<td>Data accessor</td>
<td>Queries</td>
</tr>
<tr>
<td></td>
<td>Queries</td>
<td>LMS database is the persistent data store for the LMS. The LMS consists of</td>
</tr>
<tr>
<td>Element Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LCMS database interface</td>
<td>Data accessor</td>
<td>LCMS database is the persistent data store for the LCMS. The LCMS consists of an interface that connects to its database through SQL calls</td>
</tr>
<tr>
<td>Assessment database interface</td>
<td>Data accessor</td>
<td>Assessment database is the persistent data store for the assessment system. The Assessment system consists of an interface that connects to its database through SQL calls</td>
</tr>
<tr>
<td>Collaboration database interface</td>
<td>Data accessor</td>
<td>Collaboration database is the persistent data store for the collaboration system. The Collaboration system consists of an interface that connects to its database through SQL calls</td>
</tr>
</tbody>
</table>

Table 4-5.8 Element catalogue of the Data-centered view

The relation type used in this view packet is: *attachment*. The primary presentation of this view-packet shows how the Learning Technology Components and their interfaces are attached to one another.

**Context diagram**

The context for this view packet is shown in figure 4.5-5.

**Architectural background**

A big challenge of a Shared Database solution is to come up with a suitable design for the shared database. All Learning Technology Components will have their own data schema that makes it extremely difficult to design one unified schema [HOWO04]. Another very big difficulty is that most off the shelf commercial Learning Technology Components will not work with a schema that they do not own [HOWO04]. In this thesis each Learning Technology Component will have its own database schema. Data exchange between the components is done through one of integration styles described in the previous subsections. Each of the Learning Technology Components has designed a Database Access Layer between the application and its relational database. The components can store and retrieve their persistent data by calling an appropriate method on the Database Access Layer. The Database Access Layer provides a suitable bridge to the persistent technology. Furthermore modifications to the Database Access Layer do not affect application components directly [BHS07a]. SumTotal has used .Net technology whereas Saba and Outstart have used Java technology for their design of the Data Access Layer ([OUTSa], [SABAb], [SUMTb]).

When the Learning Technology Components are hosted centrally the IT department can decide to create one Shared Database with different schemas for each database. Each component will still have its own database schema but the data exchange can be implemented on the database level. Figure 4.5.15 gives an overview of an implementation with central hosted components with a Shared Database.
4.5.6 Allocation views

Deployment view

Primary presentation

The deployment view is part of the reference architecture for e-Learning solutions and is therefore meant to be as generic as possible [Avg03]. Figure 4.5-16 show the nodes with the applications on each node that comprise an MLE. The figure also shows the protocol each of the applications use for their communication with one another.
Specific hardware requirements for Learning Technology Components that comprise the Application Server Architecture should fit as much as possible within the existing hardware infrastructure.

### Element catalogue

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Protocols</th>
<th>Hardware requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMS</td>
<td>HTTP / SOAP FTP</td>
<td>An LMS is part of a web farm with two or more LMS instances. LMS web farm requires a load-balancer that support sticky sessions. At least one LMS need a message client to receive SOAP messages from the User Management System. Each LMS instance needs an ODBC or a JDBC client to connect to its RDBMS.</td>
</tr>
<tr>
<td></td>
<td>Messages / SOAP RPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ODBC / JDBC</td>
<td></td>
</tr>
<tr>
<td>LCMS</td>
<td>HTTP / SOAP Messages / SOAP RPC ODBC / JDBC</td>
<td>An LCMS is part of a web farm with two or more LCMS instances. LCMS web farm requires a load-balancer that support sticky sessions. At least one LCMS node needs message client software to be able to receive SOAP messages from the User Management System. Each LCMS node needs an ODBC or a JDBC client to connect to its RDBMS.</td>
</tr>
<tr>
<td>Assessment System</td>
<td>HTTP / SOAP Messages / SOAP RPC ODBC / JDBC</td>
<td>An Assessment System could be part of a web farm with two or more Assessment System instances. An Assessment System could also be installed on a single node. The Assessment web farm requires a load-balancer that support sticky sessions. At least one Assessment System node needs a message client software to be able to receive SOAP messages from the User Management System. Each Assessment System node needs an ODBC or a JDBC client to connect to its RDBMS.</td>
</tr>
<tr>
<td>Collaboration System</td>
<td>HTTP /SOAP T.120 / H.323</td>
<td>A Collaboration System could be part of a web farm with two or more Collaboration System instances. A Collaboration System could also be installed on a single node. The Collaboration web farm requires a load-balancer that support sticky sessions. Each Collaboration System node needs an ODBC or a JDBC client to connect to its RDBMS.</td>
</tr>
<tr>
<td>User Management System</td>
<td>Messages / SOAP ODBC / JDBC</td>
<td>A User Management system could be part of a web farm with two or more User Management system instances.</td>
</tr>
</tbody>
</table>

Reference Architecture for e-Learning solutions

January 29, 2008; Page: 65
<table>
<thead>
<tr>
<th>Element Name</th>
<th>Protocols</th>
<th>Hardware requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR-system</td>
<td>Messages / SOAP ODBC / JDBC FTP</td>
<td>A HR- system could be part of a web farm with multiple HR-system instances. The HR-system web farm requires a load-balancer to spread the load between the nodes. The HR-system sends messages to the User Management System to update the use profiles of users. The HR-system receives FTP files with updates on skills / competencies and end-user progress on their personal development. Each HR-system node needs an ODBC or a JDBC client to connect to its RDBMS.</td>
</tr>
<tr>
<td>Portal</td>
<td>HTTP / SOAP ODBC / JDBC</td>
<td>A Portal could be part of a web farm with two or more Portal application instances. The Portal application web farm requires a load-balancer that support sticky sessions. Each Portal application node needs an ODBC or a JDBC client to connect to its RDBMS.</td>
</tr>
<tr>
<td>RDBMS</td>
<td></td>
<td>The RDBMS could be part of a cluster with different active nodes that are synchronised in which every application has its own database schema. The RDBMS could also several single nodes with a separate RDBMS per application on each node.</td>
</tr>
</tbody>
</table>

Table 4.5-9: Element catalogue deployment view

The elements in the catalogue are the Learning technology components that run on a server node in a network. The relation used in the deployment view is the ‘allocate to’ relationship that shows on which Learning Technology Component is allocated to which server node.

The hardware requirements for different Learning Technology Components and the RDBMS differ for each instance of this reference architecture. The hardware requirements for the components depend on the quality attributes such as availability, performance, scalability or cost of an architecture. The requirements mentioned in the element catalogue are based on best practices for large companies described in architecture documents of the different suppliers.
Context background
The context for this view packet is shown in figure 4.5-5.

Architectural background
There are two important principles that underpin the performance, availability and scalability of Learning Technology Components that comprise an e-Learning solution: redundancy and functional identical elements [DYLO04]. To benefit from the redundancy and functional identical elements in an e-Learning solution Load-Balanced Elements are added to balance the load across the different nodes in the network [DYLO04]. The estimation of the numbers of users a solution should support is a difficult task. The number of users or the usage of the solution could change overtime because of changing business models or a new customer base [DYLO04]. Learning Technology Components should have a Dynamically-Adjustable Configuration to be able to support changing business needs. In this thesis the reference architecture of an e-Learning solution is based on an Application Server Architecture that is implemented as a set of Load Balanced Elements which all have a Dynamically-Adjustable Configuration so that the solution can be expanded easily [DYLO04].

Each Learning Technology Component will have a database that is installed on a single server node or as a cluster of database server nodes depending on the business requirements regarding availability and cost. To reduce cost the Learning Technology Components could all use one clustered database in which each component will have its own database schema. Learning Technology Components such as SumTotal, Saba or Outstart all support solution where the application can be part of web farm with multiple instances of the application and where the database backend is a database cluster with two or more database nodes ([OUTSa], [SABAb], [SUMTd]). Figure 4-5.17 shows a solution with a web farm that consists of three LMS nodes and a database cluster with two nodes. The database cluster could also be used by other Learning Technology Components such as a LCMS that are part of an e-Learning solution.

Figure 4.5-17: Deployment view of an LMS web farm and database cluster
4.6 Evaluating the reference architecture

The evaluation of the reference architecture is based on two approaches as described in section 3.3.2.10. This section evaluates the quality attributes, which the reference architecture should fulfil, based on an experience-based assessment. The evaluation of the two business cases, of BT, in section 9 and section 10 is only available for BT LS.

Availability
According to the Module Decomposition view (see 4.5.3) there are 6 different systems that comprise the infrastructure sub-system of an MLE. These systems are all independent of one another and therefore architecture is designed in such a way that the failure of one system does not stop other systems to work. Good practice would be to disperse the different systems on separate servers, so that a crash of a server will not have an affect on the other servers and improve the availability of the whole solution ([ARS02], [DYLO04]). An LMS such as SABA or TotalLMS can also be setup as a web-farm with several independent LMS instances dispersed on different servers. The failure of one instance will have no affect on the other instances improving the availability of the solution ([SABAb], [SUMTb]). LCMS systems as Evolution or ForceTen also provide similar solutions ([OUTSa], [EEDOa]).

To be able to guarantee an availability of 99.1% adding multiple instances of systems is required. But adding redundancy to a solution has a negative impact on the cost and the maintenance of the solution so there is a trade-off made between the availability of a solution and the cost and maintenance of the solution.

Performance
The proposed Layers and Application Server Architecture of the reference architecture has a negative impact on the performance of the e-Learning solution. The negative impact of the Layers and the Application Server Architecture on the performance is caused by the extra amount of inter-communication that has to take place between the components. But the proposed Layers and an Application Server Architecture have a positive impact on the flexibility and availability of the solution ([ARS02], [DYLO04]). By separating the non-core functionality, such as mail functionality, of an e-Learning solution into Peripheral Specialist Elements has a positive impact on the overall performance of the solution [DYLO04]. The performance of the solution can also be improved if an LMS is implemented as a web-farm with multiple instances of the LMS ([SABAb], [SUMTb]). LCMS systems as Evolution or ForceTen also provide similar solutions ([OUTSa], [EEDOa]). But adding multiple instances has a negative impact on the cost and the maintenance of the solution so there is a trade-off made between the requirements for performance and cost and maintenance of the solution.

Security
The learning technology components that comprise the solution are all added to the existing infrastructure and use the security measures that are put in place by the IT departments of a corporate business. The learning components that are chosen all support LDAP integration with an organisation’s LDAP server to block unauthorised access to components [MED+06]. Because of the LDAP integration SSO can be implemented using a Front Door and Role Based Security. No extra security measures are taken to meet the properties described in section 4.3.3 other than what is available out-of-the-box in each individual
system. There are also no extra measures in place to protect the solution for to denial of service attacks, spoofing attacks or SQL injections attacks.

When it comes to the uploading capabilities of an LMS, LCMS or Assessment System extra security measures should be in place that assure that the files used for the upload only come from authorised sources [Wei05]. No extra security measures are in place for uploading content other than the security measures that is available out-of-the box in each individual system.

The security implement in the architecture provides mechanisms for the authentication and authorisation of end users. The solution could be vulnerable to attacks because no extra security measures are implemented for uploading content other than what out-of-the box available in each system.

**Usability**

Usability is a quality attribute that is difficult to evaluate in terms of a solution’s architecture because it concerns the user interface and is mostly subjectively assessed [ARS02]. Most of the learning technology components such as the LMS or LCMS provide options with which the user interface can be configured to some extend by an end-user according to his/her preferences. An LMS such as TotalLMS or SABA LMS also provide developers options to tailor the user interface according to the customer requirements. Saba and SumTotal have implemented the MVC pattern to offer multiple user interfaces without affecting the application logic ([SABAb], [SUMTb]).

The reference architecture is designed using COTS systems and because of that it is using a workflow which has been tested and used by other customers. But because the reference architecture is using COTS systems from different suppliers there could be a mismatch between the interfaces of the different systems. The latter has a negative impact on the overall usability of the solution.

**Scalability**

The Layered structure and the Application Server Architecture in itself do not improve the scalability of an e-Learning solution but they lay the foundation of a highly-scalable solution [DYLO04]. The scalability requirements could differ per learning technology component. Most of the COTS components such as an LMS or LCMS do have installation or configuration options to scale a solution for the short and long term ([EEDOa], [OUTa], [SABAb], [SUMTb]). Adding more instances of a system to a solution to fulfil new scalability requirements has a negative impact on the on the cost and the maintenance of the solution.

**Flexibility**

Flexibility is satisfied by implementing an Application Server Architecture, with the core functionality of an e-Learning solution, in a component based manner. The LMS, LCMS, Assessment System, Collaboration system all form the Application Server Architecture but each of the components could be replaced as autonomous component by another one with similar functionality ([ARS02], [DYLO04]). Peripheral Specialist Elements provide additional non-core functionality to an e-Learning solution in a very flexible way [DYLO04].

Developing an Application Server Architecture that adopts a “best-of-breed” approach has a negative impact on the overall performance of the solution because more communication between the different components is required. An Application Server Architecture that uses “best-of-breed” learning technology applications of different suppliers also has a negative
impact of the maintenance of the solution. The overall availability of the solution will be positively affected because when a component goes down it will not stop other components to work [DYLO04].

**Interoperability**

Interoperability between an LMS, an LCMS or an Assessment System is satisfied because these components all support standards such as IMS QTI, IMS Sequencing and IMS Packaging, SCORM and AICC [ARS02]. This does not mean that these components can exchange data with one another without any issues because suppliers could have different interpretations or versions these standards.

This quality attribute is also satisfied by the component based nature of the reference architecture and the different component and connector interfaces that are provided by the implementation of the File and Transfer, Messaging, Publisher-Subscriber and Remote Procedure Invocation patterns ([ARS02], [HOWO04]). The Layered structure, the Application Server Architecture, and the Peripheral Specialist Elements also make it possible to partition functionality into different components and thus promote interoperability between these components [ARS02].

**Maintainability**

The design of an Application Server Architecture that uses a “best-of-breed” approach makes the solution more flexible but has a negative impact on the maintenance of the solution. The reason for this is because systems of different suppliers are used and different integration approaches are used to integrate the systems [DYLO04].

On the other side the component based nature of the Application Server Architecture makes it easier to replace components or to upgrade a component without affecting the other components. Furthermore component based Application Server Architecture also makes it possible to isolate defects in a solution more easily [DYLO04].

To fulfil the requirements for availability, performance and scalability redundancy of systems are added to the architecture. This redundancy also has a negative impact on the maintenance of the solution because more systems need to be supported.
5 Towards a Pattern Language for e-Learning solutions

5.1 Introduction
This section describes a proposal for a pattern language for e-Learning solution. The main purpose of the proposed pattern language is to group the patterns, used for the development of a reference architecture for e-Learning solutions, together in such a way that they could be used for the development of (new) architectures.

5.2 Pattern language for e-Learning solutions
Although all of the patterns in the proposed pattern language are readily available it is difficult to connect the patterns to form a coherent and consistent pattern language for e-Learning solutions. Therefore all patterns are rewritten to place them in them more in the context of e-Learning solutions. The collection of patterns and how they are put into an e-Learning context can be found in section 5.3.
Figure 5.2-1 shows the top level patterns and how they refine the domain model in each of the views.

![Pattern language for documenting views of e-Learning solutions](image)

Domain Model
The Domain Model is the starting point of the pattern language. The Domain Model defines the scope of the reference architecture and describes the context of the views it connects to. The top level of the pattern language shows the views and how these views refine the design of Domain Model.

Module Layer View
The Module Layer View is documented by the Layer pattern. This pattern is used to describe the layers of the reference architecture.
Figure 5.2-2 shows the relationship between the Layer pattern and the Domain Objects pattern. The arrow points to the direction of the “Layer decomposition” relationship between the Layer patterns and the Domain Object pattern. The layers described by the Layer pattern are decomposed into domain objects by the Domain Objects pattern.

**Module Decomposition View**
The Module Decomposition View describes how the Layers of the Domain Model could be decomposed into smaller Domain Objects. The Domain Objects either belong to the Application Server Architecture or the Peripheral Specialist Elements. A Domain Object is part of the Application Server Architecture when it belongs to the Learning Technology Services and is part of the Peripheral Specialist Elements if it belongs to the Common Services of an MLE.

Figure 5.2-3 shows the relationship between the Domain Objects pattern with the Application Server Architecture pattern and the Peripheral Specialist Elements pattern.

**Component Interaction View**
The Component Interaction View documents the communication between the components that comprise an e-Learning solution. A component could have multiple connectors and each connector could use one of the following integration flavours: File Transfer, Messaging, Remote Procedure Invocation or Publisher-Subscriber patterns. The Component Interaction View is described in more details in the following four view packet: Component Interaction packet 1: File Transfer, Component Interaction packet 2: Messaging, Component Interaction packet 3: Remote Procedure Invocation and Component Interaction packet 4: Publisher-Subscriber.

Figure 5.2-4 shows the relationship of the Application Server Architecture pattern with integration patterns.
Component Interaction view packet 1: File Transfer
Figure 5.2-5 shows which patterns are used to design the Component Interaction view packet 1: File Transfer. The figure also shows the relationship between the File Transfer patterns and the Secure Channel pattern.

The File Transfer view packet uses the File Transfer pattern to describe how data can be sent across between components when the data in the data files have an agreed format and send at agreed and regular intervals. The Secure Channel pattern is used in the architecture of business case B as an additional security measure to transfer files with sensitive data between organisations.

Component Interaction view packet 2: Messaging
Figure 5.2-6 shows which patterns are used to document the Component Interaction view packet 2: Messaging. The figure also shows the relationship of the Messaging pattern with the Message Channel, Message Endpoint and Messages patterns.
The Messaging pattern is used to describe the messaging part of the Component Interaction view. The Messaging pattern describes how components could transfer messages across immediately, reliably and asynchronously using different formats. As shown in figure 5.2-6 a messaging system consists of a Message Channel with Message Endpoint at the end of the channel to connect the learning technology component to the channel. Messages are sent across through a Message Channel between learning technology components.

**Component Interaction view packet 3: Remote Procedure Invocation**
The Remote Procedure Invocation is a third integration style that could be used by learning technology components to exchange data with one another. In the reference architecture Remote Procedure Invocation pattern is used to integrate an LMS with an LCMS or an Assessment tool. Figure 4.5-10 shows how the remote procedures are used to provide a dynamic link to content between an LMS and an LCMS.

**Component Interaction view packet 4: Publisher-Subscriber**
Figure 5.2-7 shows which patterns are used to describe the Component Interaction view packet 4: Publisher-Subscriber. The figure also shows the relationship of the Publisher-Subscriber pattern with the Identification Authentication, Front Door and Role-based Access Control patterns

![Publisher-Subscriber pattern](image)

Figure 5.2-7: Patterns used to document a high level solution for a security mechanism

The Publisher-Subscriber pattern is used to document the Publisher-Subscriber view packet and is applied to ensure the secure access to the different components that comprise an e-Learning solution. Secure access to the components of an e-Learning solution is done by implementing a Front Door which is used as part an Identification & Authentication requirement for authentication of the end users of the solution. After the authentication of the end user a component will provide access to the end user based on the Role-based Access Control that’s implemented by the component. The Publisher-Subscriber pattern could also be used in combination with the Messaging pattern as used in the e-Learning solution of business case B where a Publisher-Subscriber Message Channel is used transfer Messages across between components.

**Data-centred View**
The data-centred view is designed using the Shared Database and the Data Access Layer patterns. The Shared Database pattern is used to discuss the challenges organisations have when it try to implement a solution with one central database for all components. The Data
Access Layer is used to define a layer between the learning technology components and their database schema.

Figure 5.2-8 shows the relationship of the Application Server Architecture with the Shared Database pattern. The figure also shows the relationship between the Shared Database pattern and the Data Access Layer pattern.

**Deployment View**

The Load-Balanced Elements and the Dynamically-Adjustable Configuration patterns are used to describe the deployment view. The Load-Balanced Elements pattern is used to address issues regarding performance and availability of an e-Learning solution. The Dynamically-Adjustable Configuration pattern is used to address the changing business needs during the life time of an e-Learning solution. Figure 5.2-9 shows the relationship of the Application Server Architecture pattern with the Load-balanced Elements and Dynamically-adjustable Configuration patterns.

**5.3 Architectural Patterns**

**5.3.1 Introduction**

This section describes the architectural patterns that are used to design views of the reference architecture for an e-Learning solution. The patterns are grouped around the aspects they address in views that document the architecture. Each pattern description describes the problem and its solution in an e-Learning context.

**5.3.2 Domain Model of an e-Learning solution**

The Domain Model is the root pattern of the pattern language that is used to develop the views of a reference architecture for e-Learning solutions ([BHS07a], [Fow03]). The Domain Model describes the scope and the vision of an e-Learning solution.

**Problem:**

Requirements and constraints describe the functionality, quality of service and deployment aspects of the systems’ underdevelopment that comprise an e-Learning solution, but they do not provide any guidelines for the development of the solution itself. Without a good insight into the scope of an e-Learning solution and its application domain the realisation could end up in a ‘big ball of mud’ that is hard to understand for the customer and a poor architectural basis to build upon.

**Solution:**
Create a model that defines and scopes the business responsibilities of an e-Learning solution and their variations where the model elements are abstractions that are meaningful in the application domain and their roles and interactions reflect the domain workflow.

The next step is to transform the Domain Model into a reference architecture. This architecture will focus on the integration of the components from the common services layer and the learning technology layer into one distributed solution. There are several groups of patterns that help transform the Domain Model into a reference architecture. The first group of patterns Layer, Domain Objects, Application Server Architecture and Peripheral Specialist Elements, deal with the system evolution of the reference architecture. The second group of patterns, Messaging and Publish and Subscribe, deal with the remote communication between applications. The third group Security patterns deal with security aspects of the components in the reference architecture.

The Layer, Domain Objects, Application Server Architecture and Peripheral Specialist Elements patterns describe the basic shape of most Internet technology systems, such as the system described by the Domain Model.

5.3.3 Layering structure of an e-Learning solution

Layered structure of the Domain Model ([BHS07a], [CBB+05], [Fow03])

Layering is one of the most common architectural structures that software designers use to break apart complex e-Learning solutions. Introducing layers into an architecture improves the independent development and evolvement of software components ([BHS07a], [CBB+05], [Fow03]).

Problem: How do we ensure that changing, replacing or adding a software component in an e-Learning solution does not force us to make major changes to the complete architecture of an e-Learning solution?

Solution: Define one or more layers for an e-Learning solution under development, where each layer has a distinct and specific responsibility. Layers ([BHS07a], [Fow03], [CBB+05]) are an important step in transforming a Domain Model ([BHS07a], [Fow03]) into an architecture but the layers alone are still too coarse-grained to support the development of a e-Learning solution. The next step is to refine the Layers into smaller, strictly separated modular parts, Domain Objects [BHS07a], which have clearly defined and scoped functionality.

5.3.4 Module Decomposition of an e-Learning solution

Domain Objects Pattern ([BHS07a], [Fow03])

When transforming the Domain Model of an e-Learning solution into a technical architecture using Layers one of the key design concerns is to decouple the Layers into self-contained and coherent applications [BHS07a]

Problem: How can we refine a Layered structure of an e-Learning solution into smaller, strictly separated modular parts which each having clearly defined and scoped responsibility?
Solution

*Provide Domain Objects that encapsulate responsibilities of components into self-contained building blocks.*

The Application Server Architecture [DYLO04] and the Peripheral Specialist Elements [DYLO04] patterns are used to divide the Domain Objects into objects that deal with the core functions and objects that deal with the non core functions of an e-Learning solution [BHS07a].

**Application Server Architecture Pattern [DYLO04]**

The Application Server Architecture pattern looks at grouping functions that belong to the core part of an e-Learning solution together and map those functions on to applications [DYLO04].

**Problem:**

*What is a good starting point for a reference architecture for e-Learning solutions?*

**Solution:**

*Adopt an Application Server Architecture that groups core functional requirements into a single solution. Internally partition the solution according to the preferred logical architecture.*

In the context of the e-Learning Domain there are several approaches possible to partition a solution into a logical architecture [Lap04]:

- Adopt a monolithic approach in which a single tightly integrated system supports most of the core functionality of the solution and is supplemented by specialised applications that address unique needs.
- Adopt a suite approach in which a group of pre-integrated applications support most of the functionality required by the enterprise.
- Adopt a “best-of-breed” approach in which applications are chosen for their fit with particular business processes and integration between the applications is managed by internally.

Dyson and Longshaw recommend the first approach because it simplifies the architecture of the solution. This first approach is trading off high-level simplicity with low-level complexity and is less flexible if it comes to adding functionality according to the customer needs [DYLO04]. In this reference architecture the second and third approach are used because functionality of an e-Learning solution is grouped into different learning technology components that are integrated into a MLE. The reference architecture does not document how these components are grouped together into one or more applications. The reference architecture only describes how some of the COTS systems such as SumTotal, Saba, or Outstart have grouped together some of the components in suite of pre-integrated applications. Instances of the reference architecture as described in case A and case B describe how components are grouped together those instances.

**Peripheral Specialist Elements Pattern [DYLO04]**

Peripheral Specialist Elements are software components that do not belong to the core part of an e-Learning solution.

**Problem:**
Should all software components be added to an e-Learning solution creating one solution that implements all system functionality?

Solution:
Separate functionality that does not belong to the ‘mainstream’ functionality from the core part of an e-Learning solution and the re-use software components from the existing infrastructure. Place this functionality at the periphery of the architecture of solution – integrated with the architecture but managed, secured and maintained independently.

By using the Applications Server Architecture pattern and the Peripheral Specialist Elements pattern the Domain Model of an e-Learning solution is divided in two parts. The Application Server Architecture pattern describes the core part of an e-Learning solution grouping learning technology functions into applications. The Peripheral Specialist Elements separates the functionality that does not belong to the mainstream functionality of an e-Learning solution by re-using components from the existing infrastructure.

5.3.5 User Interaction with an e-Learning solution
Introduction
This subsection describes the Model-View-Controller pattern as it is used on an architectural level in e-Learning solutions.

Model-View-Controller ([AVZD05], BHS07a])
Components that comprise an e-Learning solution may offer multiple user-interfaces where each user interface displays all or part of some of the application data. These components interact with one another through connectors that pass data from one to another. The interaction between the components is usually based on one of the integration styles described in the next section.

Problem
How can we develop components that comprise an e-Learning solution so that changes to application data is automatically and flexible reflected in all the different user interfaces? And how can we make changes to the user interfaces without affecting the application logic associated with the data?

Solution
Create solution in which components are divided into three parts: a Model that encapsulates some application data and logic that manipulates the data, independently from the user interfaces, one or more Views that displays parts of the data to the user, a Controller associated with each View which receives user input and translates it into a request to the Model. The user interacts strictly through the Views and their Controllers independently from the Model, which in turn will notify all interfaces about updates.

The notification mechanism that updates Views and Controllers according to the Model can be based on Publisher-Subscriber using web-services or a messaging system.

5.3.6 Component Integration Styles in an e-Learning solution
Introduction
This subsection describes the patterns that document the Component Integration Styles of an e-Learning solutions.
Component Integration Styles [HOWO04]
An enterprise has multiple applications that comprise an e-Learning solution; these applications are built independently using different languages and platforms.

Problem

How can learning technology applications that are part of an enterprise e-Learning solution be integrated so that they work together and exchange information?

There are four main approaches that deal with the fundamental challenges of integration solutions: File Transfer, Shared Database, Remote Procedure Invocation, Messaging and Publisher-Subscriber.

Solution File Transfer

Have each learning application produce files that contain the formation the other application must consume. Integrators take the responsibility of transforming files into different formats. Produce the files at regular intervals according to the nature of the business.

File Transfer enables applications to share data but it lacks timelines. To make data available more quickly than via File Transfer and to enforce a pre-defined set of data the preferred integration style is a Shared Database.

Solution Shared Database

Integrate learning applications by having them store their data in a single Shared Database, and define the schema of the database to handle all the needs of different applications.

Both File Transfer and Shared Database enable applications to share data but when a mechanism is needed for one application to invoke a function in another application, passing the data that needs to be shared and invoking the function that tells the receiving application how to process the data, the preferred integration style is Remote Procedure Invocation.

Solution Remote Procedure Invocation

Develop each learning application as a large-scale object or component with encapsulated data. Provide an interface to allow other applications to interact with the running application.

File Transfer and Shared Database enable applications to share data but not functionality. Remote Procedure Invocation enables applications to share functionality but tightly couples the applications. The Publisher-Subscriber or the Messaging integration style can be used to overcome these issues and provide functionality like File Transfer in such a way that lots of small packets can be produced and transferred asynchronously.

Solution Messaging

Use Messaging to transfer packets of data frequently, immediately, reliable and asynchronously, using customisable formats.

The concepts of Messaging are discussed in more details in section 5.3.7.

Solution Publisher-Subscriber

Use Publisher-Subscriber to exchange events frequently, immediately, reliable and asynchronously in a one-to-many configuration.

All Learning Technology Component of an e-Learning solution could be either a publisher of events or a subscriber to events. Publishers and subscribers of events are generally unaware of
one another. Subscribers are only interested in consuming events not knowing in their publishers whereas the publishers only supply events not interested in who subscribes to them. The Publisher-Subscriber pattern is a special case of the Messaging pattern where the information exchanged between components connected by Publisher-Subscriber middleware is encapsulated inside events which are realised by a messaging system. Publisher-Subscriber middleware can be implemented as web services or as a Messaging Channel [BHS07a].

5.3.7 Messaging in an e-Learning solution

Using Messaging Systems involve certain basic concepts. This subsection describes the patterns that could be used to implement the messaging concepts ([HOWO04], [BHS07a]).

Messaging Channel

In an e-Learning solution several applications need to communicate with another by using Messaging.

Problem

How do applications in an e-Learning solution communicate with another using messaging?

Solution

Connect the learning technology applications that comprise the solution by using a Message Channel, where one application writes Messages to the channel and other applications read the Messages from the channel.

A Message Channel connects a set of interacting applications allowing them to exchange Messages in a well defined and reliable manner ([HOWO04], [BHS07a]). Producers that write Messages to the channel can be sure that the receivers of the Messages are allowed to read Messages from the Message Channel and are also interested in the content of the messages. Receivers of the Messages that read the Messages can be sure that they can use and process the content from the Messages ([HOWO04], [BHS7a]).

Message Endpoint

An e-Learning solution that has applications which are communicating with each other by sending Messages via a Message Channel needs a way to write Messages to a channel or read a Message from a channel.

Problem

How do applications that comprise an e-Learning solution connect to a Message Channel to send and retrieve Messages?

Solution

Connect each application to a Message Channel using a Message Endpoint, a client of a messaging system that application can use to send or retrieve Messages.

Learning technology application that make use of a messaging system know little or nothing about messages, messaging channels or any other details of communicating with other applications through messaging. When a learning technology application want to send data across using a messaging system it passes that data to its associated Messaging Endpoint. The Messaging Endpoint takes that data, converts the data into Message that is understood by the messaging system and sends across through a Message Channel. When a learning technology application is expecting data its associated Message Endpoint extracts the data out of the Message from the Message Channel, converts it into data that that is understood by the application and passes it to the application in an appropriate format.
Messages
An e-Learning solution that has applications which are communicating via Messaging, using a Messaging Channel, needs a way to exchange data with each other.

Problem
How can applications in an e-Learning solution that are connected via Message Channel exchange information with each other?

Solution
Package the information into a Message, a data record that the messaging system can transmit through a Message Channel.

Applications in an e-Learning solution are responsible for extracting data from its database which is used by a Message Endpoint for sending Messages through a Message Channel or for importing data into its database which a Message Endpoint received from a Message Channel.

5.3.8 Security in an e-Learning solution
This sub-section describes the patterns that document the concepts of the security foundation of an e-Learning solution ([MED+06], [Wei05]).

Identification and Authentication requirements (I&A)
An I&A service for an e-Learning solution must satisfy a of requirements often set by the security department of a company. The function of I&A is to recognise an end user and validate the end user’s identity.

Problem
How can we determine a set of requirements for a I&A service and their relative importance that deal the conflicts some of these requirements have with one another and that deals with the necessary trade-offs between the conflicting requirements?

Solution
Specify a set of requirements for a specific I&A domain and determine the relative importance of each requirement.

The I&A requirements for web based applications, such as the Learning Technology components of an e-Learning solution, are often dictated by the security departments of companies. This thesis will therefore not go into the details of these requirements but only mention the solution should met the I&A requirements.

SSO is one way of implementing I&A and is used by all web based application, including the learning technology applications, to authenticate end users. A Front Door created by a LDAP system is often used to provide SSO to end users.

Front Door
An enterprise wide environment with several web based applications including Learning Technology Components need a way to authenticate end users where authenticated end users will get access to the underlying web applications.

Problem
How is SSO provided to web based Learning Technology Components or services in an e-Learning solution?

Solution
Implement a Front Door server that identifies end users and keeps track of user sessions. This server passes end user identity and session to Learning Technology Components.

A LDAP server is often implemented as Front Door for all web based applications in an enterprise wide environment including the Learning Technology Components of an e-Learning solution. In this thesis an LDAP server is part of the Peripheral Specialist Elements of a reference architecture for an e-Learning solution. Authorisation is done by the by the learning technology components by implementing Role-Based Access Control.

Secure Channels
Learning data that is sent across the Internet can be intercepted and is therefore potentially available to an eavesdropper. Extra security measures are necessary to protect sensitive learning related information that is sent across the Internet to be intercepted.

Problem
How do we ensure that learning related data being passed across a public network such as the Internet is secure in transit?

Solution
Create a secure channel for sensitive data that obscure the learning related data in transit. Exchange learning related information by setting up encrypted communication between source and destination environments. Reduce the associated overhead on the solution by using ordinary channels for non-sensitive data.

This pattern is especially important when the File Transfer pattern is used to exchange sensitive data between organisations.

Role-Based Access Control
An e-Learning solution in which we have to manage a large number of end users, information types, functionality or a large variety of resources we need a way to control access these elements.

Problem
How do we assign rights based on functions and tasks of end users?

Solution
Most organisations have a variety of job functions that require different skills and responsibilities. For security reasons end users should get rights based on their job functions or assigned tasks.

Examples of rights or permissions are the ability of an end user to create new classes of a course on an LMS, or the ability to create a new on line course in an LCMS. Learning Technology Components such as SumTotal, Saba, Outstart, ForcenTen all have functionality in place with which Administrators of those applications can define authorisation rules for roles within the application, create and delete user groups, add or remove users to user groups and assign user groups to roles.

5.3.9 Database Access in an e-Learning solution
When realising an implementation of an e-Learning solution Learning Technology Components are connected to relational databases. This section describes a pattern that provides a layer between the application and the database.
Database Access Layer
Learning Technology Components are often designed using object technology and relational databases. They use object technology because it simplifies the application design while relational databases support efficient persistence or are chosen for economical or historical reasons [BHS07a].

Problem
How can we deal with the differences between an object-oriented designed Learning Technology Component and its relational database?

Solution
Introduce a separate Database Access Layer between an Learning Technology Component and its relational database.

The advantage of introducing a Database Access Layer is that it decouples an object-oriented Learning Technology Component from the details of its database. The mappings between the objects from the component and the tables from the database are encapsulated in this layer in such a way that it appears to the component that is storing and retrieving objects instead of table entries.

5.3.10 Deployment of an e-Learning solution
The patterns in this section address the deployment aspects of an e-Learning solution [DYLO04].

Load-Balanced Elements
An e-Learning solution is an enterprise wide solution with an Application Server Architecture where availability, performance and scalability are important non-functional requirements.

Problem
How can an e-Learning solution that consists of different Learning Technology Components continue to work when one of the components become unavailable or when the load of one of the components increases beyond the capacity of a single component?

Solution
Use multiple instances of components that have the same functionality and balance the load continuously between the different instances of the component. To increase the capacity just, increase more components to the load-balanced set.

Learning Technology Components can be configured as a web farm that comprises components with the same functionality. The web farm will be accessed by a load balancer that looks after load of the individual components of the web farm.

Dynamically-Adjustable Configuration
After an e-Learning solution is deployed and has met its non-functional requirements these requirements could change over the life time of the solution.

Problem
If a Learning Technology Component used in an e-Learning solution is reaching the limit of non-functional characteristic, how can the architecture of the e-Learning solution be modified without interrupting with the solution’s operation?

Solution
Identify the parameters that fundamentally affect the non-functional characteristics of the components. For each of the parameters introduce a way for components to reload their configuration at run-time.

Important for components in an e-Learning solution is they have an option to add more component instances to a web farm to spread the load over the different component instances that comprise the web farm. This is something which can be done run-time without an interruption of the operation of a solution.

5.4 Evaluation of the pattern language

The main purpose of the proposed pattern language is to collect patterns that are used for the design of a reference architecture for e-Learning solutions. The pattern language is not a comprehensive tutorial on e-Learning solutions in general. Its main focus is on the design of the integration of the applications that comprise an e-Learning solution. The pattern language is certainly not complete; it is a start and should be seen as work in progress. As the reference architecture of e-Learning solution evolves or when more technical details are added to the views more patterns can be added or new patterns could be created to the pattern language to design the views.
6 Conclusion and recommendations for further work

The final section of this thesis deals with the answers to the research questions and revisits the problem statement. Section 2.3 of this thesis presents a problem statement and poses four research questions. Section 6.1 answers the research questions, the problem statement is discussed in section 6.2 and the last subsection 6.3 discusses recommendations for future work.

6.1 Answers to research questions

In the introduction of this thesis the following four research questions are formulated:

1. Is it possible to develop a pattern language to document a reference architecture for e-Learning solutions by re-using patterns from the different domains?
2. How is this pattern language used to document this reference architecture?
3. How can two architectures, from business cases of BT LS, be instantiated from the reference architecture and how can these two architectures be evaluated?
4. How is the reference architecture extended with specific implementation details for each of the business cases to make these implementations conform to this reference architecture?

In an attempt to answer these four research questions an approach is used that consists of four objectives:

1. Develop a reference architecture for e-Learning solutions.
   This first objective is discussed in section 4 where a reference architecture for e-Learning solution is developed. The theoretical foundation of the architecture documentation package that was created for this reference architecture is described in section 3.3.2. The views that comprise the reference architecture are designed by re-using patterns from different domains. The reference architecture is not complete but is work in progress. The most important reason for this is because of the evolvement of e-Learning solutions. Section 6.3 describes two areas which could have an impact on the evolution of the reference architecture. Another reason why the reference architecture is still being worked on is the difficulty to design a reference architecture that should be generic enough that it could be used for the design of architectures of multiple customers but also fits the description of what a reference architecture should contain ([Avg03], [Mul07]). Areas that could be improved are the business model and the technical details in the design of the views. In both cases mining into existing solutions of BT LS could provide extra information to improve the business model and technical details of design of the views [Mul07].

2. Select and describe patterns from different domains that can be used to design the relevant parts of a reference architecture for an e-Learning solution.
   Section 5.3 deals with the second objective because it describes all the patterns that are used to design the reference architecture. The patterns are selected from the following domains: Enterprise Integration Patterns [HOWO04], Security Patterns [MED+06], Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing [BHS07a], Patterns for Architecting Enterprise Solutions [DYLO04], Server Component Patterns [MAE02], Patterns for Enterprise Application Architecture [Fow03], Pattern Oriented Software Architecture: A System of Patterns [BMR+96]. The selected patterns are all modified to fit into an e-Learning context.
3. Develop a pattern language that contains the patterns that are used to document the reference architecture.

A pattern language for e-Learning solutions, the third objective, is proposed in section 5. The proposed pattern language comprises all patterns that are used to design the views of the reference architecture are described in section 4. Each pattern that is used is described in an e-Learning context in section 5.3. The patterns that comprise the pattern language are grouped together per view that they design. The pattern language is not intended to be a comprehensive guide to develop a reference architecture or architecture for e-Learning solutions.

The proposed pattern language is certainly not complete but work in progress. The first reason for this is because in this first attempt the patterns are only grouped together per view they document and a relationship is given between the different patterns. There are no new patterns created specifically for e-Learning solutions but the focus is on the re-use of existing patterns of the different domains. The patterns also do not form a tightly interwoven network that defines a process for creating architectures for e-Learning solutions but they have a loosely coupled relationship [BHS07b]. The second reason why the proposed pattern language is still being worked on is that e-Learning solutions evolve over time. Section 6.3 describes two areas which could have an impact on the evolution of the reference architecture and the patterns that are needed to design solutions for these new areas.

4. Validate the two architectures that instantiate the reference architecture and show how the reference architecture can be extended, with specific implementation details of each of the business cases.

The fourth and last objective is discussed in appendix B and C where the architectures of the two business cases that have been implemented by BT LS and have been instantiated following the steps in section 4.3.5. Case A is an example in which the initial architecture is a relatively simple architecture when it comes to the functional aspects but a complex architecture when it comes to the infrastructure. The architecture in business case A is implemented in such a way that it can grow to a full service oriented model with different learning services that are embedded in the infrastructure of customer A. The second case B is an example in which the e-Learning solution is embedded in the infrastructure of customer B and its partners.

Both cases show that it is possible to instantiate the reference architecture. The views of both architectures are designed by using patterns from the proposed pattern language in section 6. Both architectures are also evaluated by using the quality attributes described in section 4. The most important conclusion that can be drawn from the evaluation of both architectures is that it is critical to choose learning technology components that meet the functional requirements but also the quality attributes of the architecture.

By achieving all the research objectives, the following answers to the research questions may be provided:

1. Answer to research question one. In order to answer this question a reference architecture is developed of which the are were designed by patterns. After the reference architecture was completed a pattern language was proposed. This is covered by objective one and two.

2. Answer to research question two. The patterns from the pattern language are used to define the scope of the reference architecture and to design the different views of the reference architecture. This is covered by objective three.
3. Answer to research question three. Steps are described in the reference architecture how an instance should be created. The instances of the reference architecture were also evaluated by case A and case B in the appendix. This is covered by objective four.

4. Answer to research question four. The way in which the two architectures of the business cases extend the reference architecture is also covered by objective four and is described in the appendix.

6.2 Problem statement

The problem statement formulated for this thesis is:

This thesis investigates if a reference architecture for e-Learning solutions can be developed and its views can be designed by re-using architectural patterns. This thesis also investigates if a pattern language can be developed for e-Learning solutions.

Taken the answers on the research question as described in the previous section into account a conclusion may be drawn that it is possible to develop a reference architecture for e-Learning solutions. Architectural patterns were used to design the views of the reference architecture. The reference architecture is still work in progress due to the evolvement of e-Learning solutions and some improvements were suggested to the reference architecture. The pattern language that is proposed is a first attempt to develop a pattern language for e-Learning solutions. In this first attempt patterns are grouped based on the view they document and a loosely relationship between patterns was given. Suggestions are given to improve the pattern language to bring it more in line with what the purpose of pattern language is according to the literature. The pattern language will also evolve over time due the evolution of MLEs.

The two business cases demonstrate that the reference architecture can be instantiated and evaluated. This means that BT LS can instantiate this reference architecture and therefore it can be used by BT LS to design new architectures for e-Learning solutions.

6.3 Recommendation for future work

Service Oriented Architecture approach

There is a shift in which learning technology applications such as an LMS or LCMS move away from being a monolithic application towards more flexible components that expose services accessible to other applications via a loosely coupled standards-based interface. Suppliers of learning technology applications such as SumTotal, SABA or Outstart add web-services to their application to make the integration between their applications and other learning technology components easier. Besides these enterprise applications there also exist smaller teaching applications, simulation applications or applications for serious gaming that are developed in isolation for a specific purpose and audience without regard to standards. The adoption of service oriented approaches should enable these applications to integrate with an enterprise wide e-Learning solution. A fully service oriented approach is still far from the norm. In this thesis some initial steps have already been made towards a more service oriented approach, further research is necessary to understand the consequence when the reference architecture described in this thesis evolves to a SOA. The benefits and drawbacks of this approach need to be addressed from a number of different perspectives, including performance, availability, user experience, security and development and maintenance cost ([BMMW04], [Wel07]).
Reusable content
The current approaches for finding content, reusing it and repurposing it are supported by a wide range of standards such as IMS Content Packaging, SCORM and IEEE LOM. Content is typically stored in file systems or content management systems that are coupled with learning delivery by an LMS operated for a single organisation. The reference architecture described in this thesis deals with this kind of e-Learning solutions. However the requirement for interoperable repositories and learning management systems in which a user can easily search for content, a user is able to find content in a context and a user’s search result only returns what they need indicates that a more flexible model for of learning content is needed ([BMMW04], [Wel07]). One example of such a model is the Content Object Repository Discovery and Registration / Resolution Architecture (CORDRA). CORDRA is designed to be an enabling model to bridge the world of learning content management and delivery and the world of content repositories and digital libraries. CORDRA is based on existing standards and specifications and is designed to fit within the existing environments and infrastructure of an organisation [LSAL04]. Key components of an environment that is based on CORDRA are:

- **Content repositories**: Local repositories for storing learning content.
- **Identifier Infrastructure**: Infrastructure for object identification, registration and resolution.
- **Common services**: Technical and administrative services such as authentication services, digital right management services etc.
- **Applications**: Applications and services used to manage and deliver learning content and content objects to the end user.

Further investigation is necessary to understand how models such as CORDRA can enhance the reference architecture and the pattern language described in this thesis.
# References

<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[AGR04]</td>
<td>Avgeriou, P; Guelfi, N; Razavi, R; A Pattern Language for Documenting Software Architectures, proceedings of the 9th European Pattern Languages of Programming (EuroPLOP) 7-11 July 2004, Irsee, Germany</td>
</tr>
<tr>
<td>[AGR04]</td>
<td>Avgeriou, P; Guelfi, N; Razavi, R; A Pattern Language for Documenting Software Architecture, proceedings of the 9th European Pattern Languages of Programming (EuroPLOP) 7-11 July 2004, Irsee, Germany</td>
</tr>
<tr>
<td>[Ale79]</td>
<td>Alexander, C; The Timeless Way of Building, Oxford University Press, 1979</td>
</tr>
<tr>
<td>[ARS02]</td>
<td>Avgeriou, P; Guelfi, N; Razavi, R; A Pattern Language for Documenting Software Architectures, proceedings of the 9th European Pattern Languages of Programming (EuroPLOP) 7-11 July 2004, Irsee, Germany</td>
</tr>
<tr>
<td>[ARS03]</td>
<td>Avgeriou, P; Retalis, S; Skordalakis, M; Building Quality into Learning Management Systems and Architecture-Centric Approach, International Workshop on Conceptual Modelling Quality (IWCMQ02), Tampere, Finland, October 11, 2002</td>
</tr>
<tr>
<td>[AVZD05]</td>
<td>Avgeriou, P; Zdun, U; Architectural Patterns Revisited – A Pattern Language; The European Pattern Language of Programming (EuroPLOP) 6th – 10th July 2005, Irsee, Germany</td>
</tr>
<tr>
<td>[BHS07a]</td>
<td>Buschmann, F; Henney, K; Schmidt, D; Pattern-Oriented Software Architecture, A Pattern Language for Distributing Computing, Volume 4; J.Wiley &amp; Sons Ltd., 2007</td>
</tr>
<tr>
<td>[BHS07b]</td>
<td>Buschmann, F; Henney, K; Schmidt, D; Pattern-Oriented Software Architecture, A Pattern Language for Distributing Computing, Volume 5; J.Wiley &amp; Sons Ltd., 2007</td>
</tr>
<tr>
<td>[BMR+96]</td>
<td>Buschmann, F; Meunier, R; Rohner, H; Sommerlad, P; Stal, M; Pattern oriented Software Architecture – A system of Patterns, J.Wiley &amp; Sons Ltd., 1996</td>
</tr>
<tr>
<td>[BOHO06]</td>
<td>Boucke, N; Holvoet T; Relating Architectural Views with Architectural Concerns, Proceedings of the 2006 international workshop on Early Aspects At ICSE (Shanghai, China, May 21, 2006), 2006</td>
</tr>
<tr>
<td>[Bos00]</td>
<td>Bosch, J; Design and Use of Software Architecture: Adopting and Evolving a Product-Line Approach; Addison-Wesley; 2000</td>
</tr>
<tr>
<td>[BTLS05]</td>
<td>e-Learning brief: Release the potential of you people with blended learning; internal BT LS document; 2005</td>
</tr>
<tr>
<td>[CBB+05]</td>
<td>Clemens, P; Bachmann, F; Bass, L; Garlan, D; Ivers, R; Nord, R; Stafford, J; Documenting Software Architecture – Views and beyond; SEI Series in Software Engineering, Addison – Wesley, 7th printing, 2005</td>
</tr>
<tr>
<td>[DYLO04]</td>
<td>Dyson, P; Longshaw, A; Architecting Enterprise Solutions, J.Wiley &amp; Sons Ltd., 2004</td>
</tr>
<tr>
<td>[EEDOa]</td>
<td>ForceTen Architecture version 4.1; whitepaper from EEDO, download from <a href="http://www.eedo.com">www.eedo.com</a></td>
</tr>
<tr>
<td>[Fow03]</td>
<td>Fowler, M; Patterns Enterprise Application Architecture; Addison-Wesley, 2003</td>
</tr>
<tr>
<td>[GUINa]</td>
<td>Guinti Interactive user guide, Learn Exact; <a href="http://www.eedo.com">Learn Exact user manual version 4.0;</a> Guinti Interactive user guide</td>
</tr>
<tr>
<td>[HOWO04]</td>
<td>Hohpe, G; Woolf, B; Enterprise Integration Patterns, Designing, Building and Deploying Messaging</td>
</tr>
</tbody>
</table>
Solutions: Addison-Wesley, 2004

[IBMa] Learning infrastructure: architecting a formal and informal learning environment; whitepaper from IBM Learning Solutions; 2006; downloaded from www.ibm.com/learning


[IBM06] Learning infrastructure: architecting a formal and informal learning environment; IBM Learning Solutions Executive Brief; 2006


[JISC04] Review of JISC funded Managed Learning Environment (MLE) activity; JISC report, downloaded from www.jisc.ac.uk ; 2004

[JISC05] Case Studies of Managed Learning Environments in Further Education; JISC report, downloaded from www.jisc.ac.uk , 2005


[Kru95] Kruchten, P; The 4+1 view model of architecture; IEEE software, 12(6): 42-50, 1995

[Lap04] Lapkin, A; A User’s Guide to Architectural Patterns; Gartner Research Paper; ID Number: G00124049, 2004


[MAE02] Völter, M; Scmid, A; Wolff, W; Server Component Patterns, Component Infrastructures Illustrated with EJB; J.Wiley & Sons Ltd., 2002

[MED+06] Schumacher, M; Fernandez-Buglioni, E; Hybertson, D; Buschmann, F; Sommerlead, P; Security Patterns, Integrating Security and Systems Engineering, J.Wiley & Sons Ltd., 2006


[Mul07] Muller, G; A Reference Architecture Primer; whitepaper from Gaudi project; downloaded from www.gaudisite.nl; 2007

[OUTSa] Évolution Java Architecture; Outstart whitepaper, downloaded from www.outstart.com

[OUTSb] AICC integration with Docent LMS, Outstart whitepaper, downloaded from www.outstart.com

[QMPa] Shepard, E; Godwin, J; Assessment through the Learning Process; Question Mark Perception whitepaper; downloaded from www.qmp.com


[Pau02] Paulsen, M; Online Education Systems: Discussion and Definition of Terms; NKI Distance Education; white paper downloaded from www.nettskolen.com, 2002

[REAV02] Retalis, S; Avgeriou, P; Modelling Web-based Instructional Systems, Journal of Information Technology Education, Volume 1 N0. 1, 2002

[Ris98] Rising, L; The Patterns Handbook: Techniques, Strategies and Applications; Cambridge University Press, 1998

[Ree02] Reed, P; Reference Architecture: The best of best practices; whitepaper from IBM; 2002; downloaded from www.ibm.com


[Ros06] Rosenberg, S; Beyond e-Learning, Pfeiffer, 2006

[Ros06a] Rosen, M; What is a reference architecture; document from Cutter; downloaded from www.cutter.com

[SABAa] Saba Enterprise 5.3; System Administration Guide; a Saba guide, G-SAG-SF53000-0

[SABAb] Saba Enterprise 5.3, Architecture fundamentals, a Saba guide, SL005x-ARCFUN-12.p

[SABAc] Saba Enterprise 5.3, Data migration and synchronisation, a Saba guide, SE005x-DATMIG-12.p
<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SABAa</td>
<td>Saba VLE Connector API 1.0, Developers Guide, Saba whitepaper downloaded from <a href="http://www.saba.com">www.saba.com</a></td>
</tr>
<tr>
<td>SABAe</td>
<td>Saba Enterprise 5.3; Saba web-services developers guide, whitepaper from Saba downloaded from <a href="http://www.saba.com">www.saba.com</a></td>
</tr>
<tr>
<td>SUMTa</td>
<td>SumTotal LMS 7.6; User and Managing TotalLMS 7.6; SumTotal user guide downloaded from <a href="http://www.SumTotal.com">www.SumTotal.com</a></td>
</tr>
<tr>
<td>SUMTb</td>
<td>SumTotal LMS 7.6; Configuring and Administering TotalLMS 7.6; SumTotal user guide downloaded from <a href="http://www.SumTotal.com">www.SumTotal.com</a></td>
</tr>
<tr>
<td>SUMTd</td>
<td>SumTotal TotalLMS 7.1, SumTotal web-services 7.0; SumTotal whitepaper; downloaded from <a href="http://www.SumTotal.com">www.SumTotal.com</a></td>
</tr>
<tr>
<td>SUMTc</td>
<td>Frederico, J; SumTotal Enterprise Suite, Product Architecture whitepaper; SumTotal whitepaper; downloaded from <a href="http://www.SumTotal.com">www.SumTotal.com</a></td>
</tr>
<tr>
<td>SUMTf</td>
<td>SumTotal TotalLMS 7.1, SumTotal web-services 7.0; SumTotal whitepaper; downloaded from <a href="http://www.SumTotal.com">www.SumTotal.com</a></td>
</tr>
<tr>
<td>SUMTe</td>
<td>Working with the Data Synchronisation Tool on TotalLMS 7.6; SumTotal whitepaper; downloaded from <a href="http://www.SumTotal.com">www.SumTotal.com</a></td>
</tr>
<tr>
<td>SUMTe</td>
<td>Thompson, M; Docent Enterprise System Architecture; Docent Technology Paper; SumTotal whitepaper; downloaded from <a href="http://www.SumTotal.com">www.SumTotal.com</a></td>
</tr>
<tr>
<td>SUN03</td>
<td>SUN Microsystems; E-Learning Framework, Technical Whitepaper, SUN, 2003</td>
</tr>
<tr>
<td>UKEU02</td>
<td>UK e-Universities Worldwide, Principles and Practice in e-Learning platform architecture, UKeU, 2002</td>
</tr>
<tr>
<td>VOC+06</td>
<td>Vanthournout, D; Olson, K; Ceisal, J; White, A; Waddington, T; Barfield, T; Desai, S; Mindrum, C; Return on Learning: Training for High Performance at Accenture; Agate; 2006</td>
</tr>
<tr>
<td>Wei05</td>
<td>Weippl, E. R; Security in E-Learning, Springer, 2005</td>
</tr>
<tr>
<td>Wel07</td>
<td>Weller, M; Virtual Learning Environments, Using, choosing and developing your VLE; RoutledgeFalmer, 2007</td>
</tr>
<tr>
<td>WOJP04</td>
<td>Wilson, S; Oliver, B; Jeyes, S; Powell, A; A Technical Framework to Support e-Learning; JISC paper; 2004; downloaded from <a href="http://www.jisc.org">www.jisc.org</a></td>
</tr>
</tbody>
</table>

Supplier reference:
- CENT: Centra web site at: www.centra.com
- EEDO: Eedo web site at: www.eedu.com
- GUIN: Guinti Interactive web site at: www.learnexact.com
- MOOD: Moodle web site at: www.moodle.org
- OUTS: Outstart web site at: www.outstart.com
- QMP: Question Mark Perception web site at: www.qmp.com
- SABA: Saba web site at: www.saba.com
- SUMT: SumTotal web site at: www.SumTotal.com
- WEBX: WebEx web site at: www.webex.com
8 Appendix A

8.1 List of acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADL</td>
<td>Advanced Distributed Learning</td>
</tr>
<tr>
<td>AICC</td>
<td>Aviation Industry CBT Committee</td>
</tr>
<tr>
<td>CORDRA</td>
<td>Content Object Repository Discovery and Registration / Resolution Architecture</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Off The Shelf components</td>
</tr>
<tr>
<td>ECDL</td>
<td>European Computer Driving License</td>
</tr>
<tr>
<td>HR</td>
<td>Human Resource system</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IMS</td>
<td>Instructional Management System Global Learning Consortium</td>
</tr>
<tr>
<td>JISC</td>
<td>Joint Information Systems Committee</td>
</tr>
<tr>
<td>LCMS</td>
<td>Learning Content Management System</td>
</tr>
<tr>
<td>LDAP</td>
<td>Lightweight Directory Access Protocol</td>
</tr>
<tr>
<td>LMS</td>
<td>Learning Management System</td>
</tr>
<tr>
<td>LTSA</td>
<td>Learning Technology Systems Architecture</td>
</tr>
<tr>
<td>MLE</td>
<td>Managed Learning Environment</td>
</tr>
<tr>
<td>ODBC / JDBC</td>
<td>Open Data Base Connectivity / Java Data Base Connectivity</td>
</tr>
<tr>
<td>QTI</td>
<td>IMS Question and Test Interoperability</td>
</tr>
<tr>
<td>RDBMS</td>
<td>Relational Database Management System</td>
</tr>
<tr>
<td>RLO</td>
<td>Re-usable Learning Object</td>
</tr>
<tr>
<td>RPC</td>
<td>Remote Procedure Call</td>
</tr>
<tr>
<td>SCORM</td>
<td>Sharable Content Object Model</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
<tr>
<td>SSO</td>
<td>Single Sign On</td>
</tr>
<tr>
<td>VLE</td>
<td>Virtual Learning Environment</td>
</tr>
</tbody>
</table>

8.2 Glossary of terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AICC</td>
<td>AICC has and is developing standards for interoperability of computer based and computer managed training material across multiple industries.</td>
</tr>
<tr>
<td>Assessment System</td>
<td>A software application to evaluate learner’s skills or knowledge in a particular subject area.</td>
</tr>
<tr>
<td>Authoring Tool</td>
<td>Software application used to produce electronic learning content.</td>
</tr>
<tr>
<td>Collaboration</td>
<td>Collaboration is the part of an e-Learning solution that is responsible for all synchronous and asynchronous interactions between end-users of the solution.</td>
</tr>
<tr>
<td>e-Learning</td>
<td>Learning where the Internet plays an important role in the delivery, support, administration and assessment of learning.</td>
</tr>
<tr>
<td>Event Management</td>
<td>Event Management provides calendar, scheduling and reminder functionality to all end-users of an e-Learning solution.</td>
</tr>
<tr>
<td>System</td>
<td>System discipline and in particular its basic HR activities and processes with the information technology field.</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LCMS</td>
<td>A software application (or set of applications) that manages the creation, storage, use and re-use of learning content.</td>
</tr>
<tr>
<td>LMS</td>
<td>A software application that registers users, tracks courses in a catalogue, records data from learners and provides reports to management. An LMS usually does not include its own authoring capabilities; instead it focuses on managing courses by a variety of other sources.</td>
</tr>
<tr>
<td>(Learning) Portal</td>
<td>Any website that offers learners or organisations consolidated access to learning and training resources from multiple sources.</td>
</tr>
<tr>
<td>Messaging System</td>
<td>Messaging systems use Messaging as integration style between different systems. Messaging is used to transfer packets of data frequently, immediately, reliable and asynchronously, using customisable formats.</td>
</tr>
<tr>
<td>MLE</td>
<td>A whole range of learning technology systems and processes of a corporate business or a educational institution that contribute directly or indirectly to learning and management of that learning.</td>
</tr>
<tr>
<td>Reference Architecture</td>
<td>A reference architecture is a software architecture at a higher level of abstraction than a software architecture.</td>
</tr>
<tr>
<td>Service Oriented Architecture</td>
<td>A Service Oriented Architecture (SOA) is a software architecture where flexible components expose services accessible to applications via loosely coupled standard-based interfaces.</td>
</tr>
<tr>
<td>Software Architecture</td>
<td>A software architecture is the structure or structures of a system, which comprises software elements, the externally visible properties of those elements, and the relation between them.</td>
</tr>
<tr>
<td>SCORM</td>
<td>A set of specifications that, when applied to course content, produces small, re-usable learning objects. A result of ADL’s SCORM initiative, SCORM compliant courseware elements can easily merged with other compliant elements to produce a highly modular repository of courseware material.</td>
</tr>
<tr>
<td>User Management</td>
<td>User Management is the part of an e-Learning solution that manages users and groups. User Management is also the backbone for the authentication, authorisation and entitlement of end-user of an e-Learning solution.</td>
</tr>
<tr>
<td>Virtual Classroom</td>
<td>A software application that delivers scheduled offerings to multiple locations via a network solution.</td>
</tr>
<tr>
<td>VLE</td>
<td>Software components in which learners and instructors participate in on-line interactions of various kinds, including on-line learning.</td>
</tr>
</tbody>
</table>